# Delivery Company DataBase System

NAME: NISCHAL A ROLL NUMBER: 1801CS33

```
mysql> show tables;
+-------------------+
| Tables_in_project |
+-------------------+
| bill              |
| bill_type         |
| customer          |
| delivered         |
| delivery          |
| delivery_status   |
| hazardous         |
| international     |
| recepient         |
| regular           |
+-------------------+
10 rows in set (0.00 sec)

mysql>
```

Tables present

1) Customer

| Attr Name | Attr Type |
|---|---|
| cid | Int (primary key) |
| Cname | Varchar (25) |
| Street address | Varchar (50) |
| City | Varchar (25) |
| Country | Varchar (25) |
| Phone number | Varchar (20) |
| Num of deliveries | Int – default 0 |
| Is company | int |

2) Receipient

| Attr Name | Attr Type |
|---|---|
| rid | Int (primary key) |
| rname | Varchar (25) |
| Street address | Varchar (50) |
| City | Varchar (25) |
| Country | Varchar (25) |
| Phone number | Varchar (20) |
| Num of deliveries | Int – default 0 |

3) Delivery

| Attr Name | Attr Type |
|---|---|
| tid | Int (primary key) |
| cid | Int (fk) |
| rid | Int (fk) |
| type | Varchar (15) |
| Initiated Date | date |

4) International

| Attr Name | Attr Type |
|---|---|
| tid | Int (foreign key, pk) |
| type | Varchar (10) |
| cost | Int |

| | |
|---|---|
| Promised date | date |

4) Regular

| Attr Name | Attr Type |
|---|---|
| tid | Int (foreign key, pk) |
| type | Varchar (15) |
| cost | Int |
| Promised date | Date |

4) Hazardous

| Attr Name | Attr Type |
|---|---|
| tid | Int (foreign key, pk) |
| type | Varchar (15) |
| cost | Int |
| Promised date | date |

5) Bill

| Attr Name | Attr Type |
|---|---|
| tid | Int (pk) |
| cid | Int (fk) |
| rid | Int (fk) |
| type | Varchar (15) fk |
| Amount due | int |
| Pay Status | Varchar (3) |
| Date paid | date |

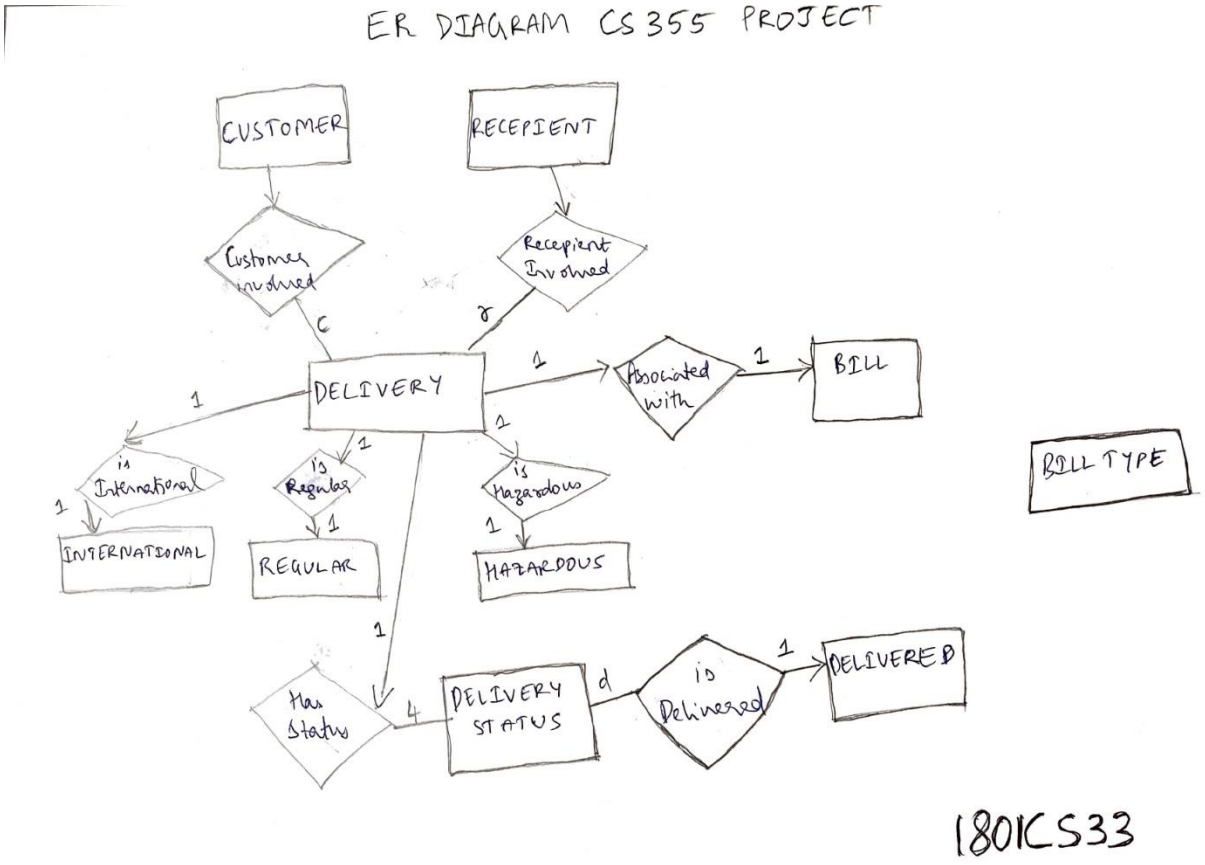
6) Delivery Status

| Attr Name | Attr Type |
|---|---|
| tid | Int (fk) |
| Date_t | date |
| status | Varchar (20) |
| Transport mode | Varchar (6) |
| Transport number | Char (5) |
| location | Varchar (20) |

7) Delivered

| Attr Name | Attr Type |
|---|---|
| tid | Int (fk) |
| Date promised | date |
| Date delivered | date |
| Transport number | Char (5) |
| location | Varchar (20) |

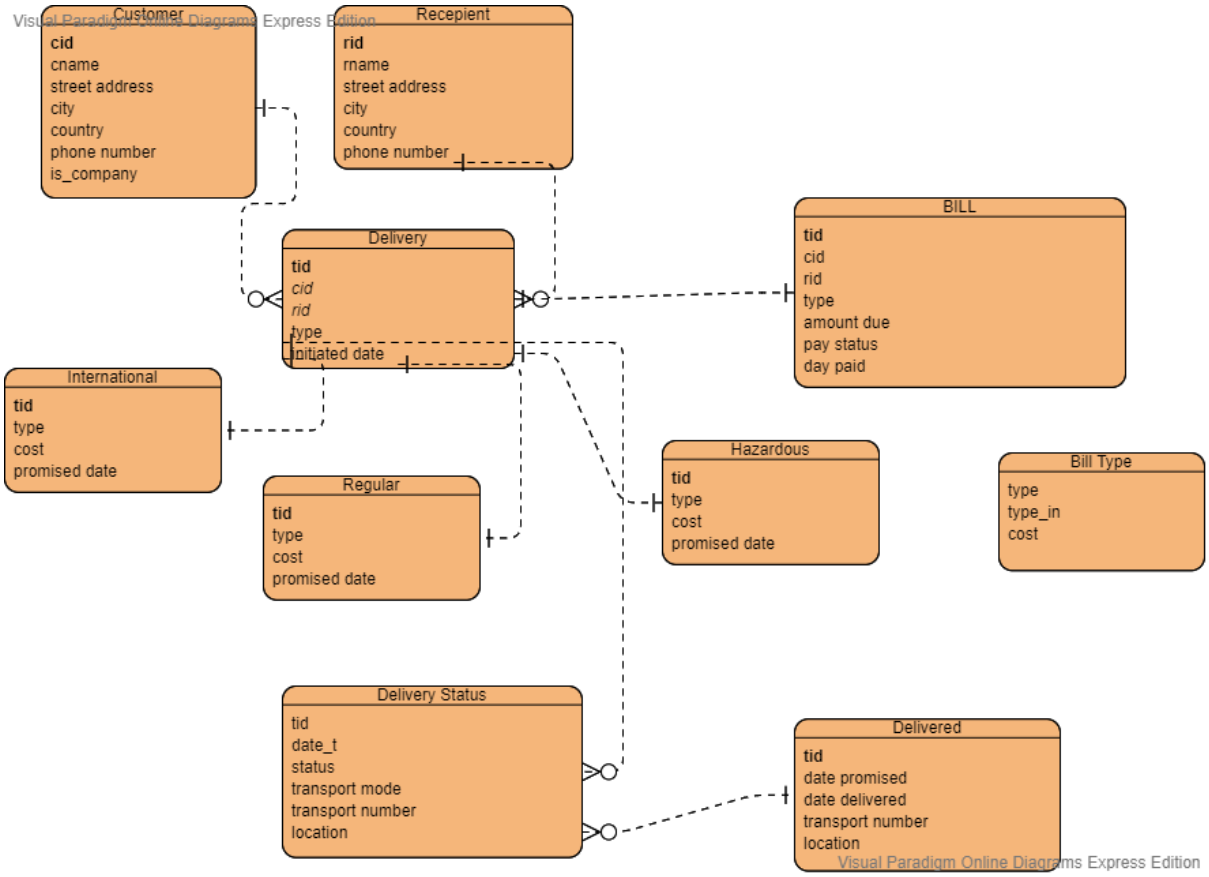
ENTITY RELATIONSHIP DIAGRAM




| Relation | Entities Involved | Type |
|---|---|---|
| Customer Involved | Customer, delivery | One to many |
| Recipient Involved | Recipient, delivery | One to many |
| Associated with | Delivery, bill | One to one |
| Is international | Delivery, international | One to one |
| Is regular | Delivery, regular | One to one |

| | | |
|---|---|---|
| Is hazardous | Delivery, hazardous | One to one |
| Has status | Delivery, delivery_status | One to many |
| Is delivered | Delivery_status, delivered | many to one |

RELATIONAL SCHEMA




The above relation schema shows all the tables and the relations between them.

The standard convention of using big arrow is used to denoted many side of the relationship and using the small arrow i.e. straight line perpendicular to the connecting line is used to show the one side of the relationship.

Sequence of Events

a) Create all tables

1) Populate the customer table with relevant entries

2) Populate the recipient table with relevant entries

3) Populate delivery table accordingly

4) Trigger on after insert delivery table to insert in either of international, normal, or hazardous tables (fill randomly)

Trigger on after insert in delivery table for filling status table

5) trigger on after insert in either of international, normal, or hazardous tables to insert in the BILL table

6) Populate delivery status table with relevant entries

7) trigger on after insert delivery status to insert into delivered table if status is delivered.

8) trigger of insert in delivered to update delivered count for customer and recipient

8) Perform all basic required queries.

9) Add additional queries

TOTAL DATA IN THE DATABASE

| Table Name | Number of Rows |
|---|---|
| customer | 401 |
| recipient | 300 |
| delivery | 200 |
| international | 58 |
| hazardous | 79 |
| regular | 63 |
| delivery_status | 375 |
| delivered | 29 |
| bill_type | 7 |
| bill | 200 |

Total Number of Rows in the entire Database is 1712

Simple queries

➔ Create database project;
➔ Use project;

```
/*Creating table customer*/

create table customer(

cid int primary key,

cname varchar (25),

street_address varchar (50),

city varchar (25),

country varchar (25),

phone_number varchar(20),

num_deliveries int,

is_company int);


/*Loading data from the csv file into the table*/

LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server
8.0\\Uploads\\customer_data.csv' INTO TABLE customer FIELDS TERMINATED
BY ',' LINES TERMINATED BY '\n';


/*Creating the table recipient*/

create table recepient(

rid int primary key,

rname varchar (25),

street_address varchar (50),

city varchar (25),

country varchar (25),

phone_number varchar(20),
```

```
num_deliveries int);

/*Loading the data from the file*/
LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\recepient_data.csv' INTO TABLE recepient FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n';

/*Creating the table delivery*/
create table delivery(
tid int primary key,
cid int,
rid int,
type varchar (15),
initiated_date date,
constraint d1 foreign key (cid)
references customer (cid),
constraint d2 foreign key (rid)
references recepient (rid));

/*Creating the table international*/
->create table international(
tid int primary key,
type varchar (15),
cost int,
promised_date date,
constraint i1 foreign key (tid)
references delivery (tid));
```

```
/*Creating the table regular*/
->create table regular(
tid int primary key,
type varchar (15),
cost int,
promised_date date,
constraint r1 foreign key (tid)
references delivery (tid));

/*Creating the table hazardous*/
->create table hazardous(
tid int primary key,
type varchar (15),
cost int,
promised_date date,
constraint h1 foreign key (tid)
references delivery (tid));

/*Creating the table bill*/
create table bill(
tid int primary key,
cid int,
rid int,
type varchar (15),
amount_due int,
pay_status varchar (3),
```

```
paid_date date,
constraint b1 foreign key (tid)
references delivery (tid));

/*Creating the table delivery_status*/
create table delivery_status(
tid int primary key,
date_t date,
status varchar (20),
transport_mode varchar (6),
transport_number char (5),
location varchar (20),
constraint ds1 foreign key (tid)
references delivery (tid));

/*Creating the table delivered*/
create table delivered(
tid int primary key,
promised_date date,
delivered_date date,
transport_number char (5),
location varchar (20),
constraint dd1 foreign key (tid)
references delivery (tid));
delimiter $$
delimiter $$
```

```
/*Creating trigger on delivery after insert. Based on the type of package i.e international, hazardous, regular, the subsequent tables are filled with random values (carefully generated)*/

create trigger after_delivery_insert

after insert

on delivery for each row

begin

declare type_t varchar (20);

select type

into type_t

from delivery

where tid= new.tid;

if type_t = 'international' then

call insertInternational(new.tid);

elseif type_t = 'regular' then

call insertRegular(new.tid);

elseif type_t = 'hazardous' then

call insertHazardous(new.tid);

end if;

call initiateStatus(new.tid);

end $$

delimiter ;
```

```
/*Procedure for initiating the delivery i.e add initiate status into the
delivery_status table as soon as a delivery is logged in*/
delimiter $$
create procedure initiateStatus(in tid_t int)
begin
declare initiate date;
declare location varchar (25);
declare cid_t int;
declare type_t varchar (20);
select type into type_t
from delivery where tid= tid_t;
select initiated_date into initiate
from delivery where tid = tid_t;
select cid into cid_t
from delivery where tid = tid_t;
select city into location
from customer where cid = cid_t;
if type_t = "international" then
insert into delivery_status
values(tid_t, initiate, "Initiated", "Truck", "T0000", location);
elseif type_t = "regular" then
insert into delivery_status
values(tid_t, initiate, "Initiated", "Truck", "T0001", location);
elseif type_t = "hazardous" then
insert into delivery_status
values(tid_t, initiate, "Initiated", "Truck", "T0010", location);
end if;
```

```
end $$
delimiter ;


/*Procedure for inserting values into the international table. It takes the tid and fills in other details like the “fast” and “standard” option randomly and sets the price accordingly*/
delimiter $$
create procedure insertInternational(in tid_t int)
begin
declare random_n int;
declare initiate date;
declare new_date date;
set random_n = floor(rand()*(2)+1);
select initiated_date
into initiate
from delivery
where tid = tid_t;
if random_n = 1 then
set new_date = adddate(initiate, interval 3 day);
insert into international
values(tid_t, "fast", 1500, new_date);
elseif random_n = 2 then
set new_date = adddate(initiate, interval 5 day);
insert into international
values(tid_t, "standard", 1200, new_date);
end if;
end $$
```

```
delimiter ;


/*Procedure for inserting values into the regular table. It takes the tid and fills in other details like the “small envelope”, “large box” and “small box” option randomly and sets the price accordingly*/

delimiter $$

create procedure insertRegular(in tid_t int)

begin

declare random_n int;

declare initiate date;

declare new_date date;

set random_n = floor(rand()*(3)+1);

select initiated_date

into initiate

from delivery

where tid = tid_t;

if random_n = 1 then

set new_date = adddate(initiate, interval 2 day);

insert into regular

values(tid_t, "small envelope", 200, new_date);

elseif random_n = 2 then

set new_date = adddate(initiate, interval 2 day);

insert into regular

values(tid_t, "small box", 600, new_date);

elseif random_n = 3 then

set new_date = adddate(initiate, interval 3 day);

insert into regular
```

```
values(tid_t, "large box", 800, new_date);
end if;
end $$
delimiter ;
```

/*Procedure for inserting values into the hazardous table. It takes the tid and fills in other details like the “chemicals” and “harmless” option randomly and sets the price accordingly*/

```
delimiter $$
create procedure insertHazardous(in tid_t int)
begin
declare random_n int;
declare initiate date;
declare new_date date;
set random_n = floor(rand()*(2)+1);
select initiated_date
into initiate
from delivery
where tid = tid_t;
if random_n = 1 then
set new_date = adddate(initiate, interval 3 day);
insert into hazardous
values(tid_t, "chemicals", 1000, new_date);
elseif random_n = 2 then
set new_date = adddate(initiate, interval 2 day);
insert into hazardous
values(tid_t, "harmless", 800, new_date);
```

```
end if;
end $$
delimiter ;
```

/*Trigger on after insertion on international to update the bill value. All details like tid, cid, rid are procured from different relevant tables. The promised date and initiated date are calculated and inserted in the BILL table along with the cost calculated. Pay status is set as “yes” (meaning paid) to normal customers and “no” (meaning not paid) to companies which have a pact with the delivery agency (and hence pay monthly)*/

```
delimiter $$
create trigger after_international_insert
after insert
on international for each row
begin
declare cid_t int;
declare rid_t int;
declare type_t varchar (20);
declare customer_t int;
declare date_paid_t date;
select type into type_t
from delivery where tid= new.tid;
select cid into cid_t
from delivery where tid = new.tid;
select rid into rid_t
from delivery where tid = new.tid;
select is_company into customer_t
from customer where cid = cid_t;
```

```
select initiated_date into date_paid_t
from delivery where tid = new.tid;
if customer_t = 0 then
insert into bill
values(new.tid, cid_t, rid_t, type_t, new.cost, "yes", date_paid_t);
elseif customer_t = 1 then
insert into bill
values(new.tid, cid_t, rid_t, type_t, new.cost, "no", NULL);
end if;
end $$
delimiter ;
```

/*Trigger on after insertion on international to update the bill value. All details like tid, cid, rid are procured from different relevant tables. The promised date and initiated date are calculated and inserted in the BILL table along with the cost calculated. Pay status is set as “yes” (meaning paid) to normal customers and “no” (meaning not paid) to companies which have a pact with the delivery agency (and hence pay monthly)*/

```
delimiter $$
create trigger after_regular_insert
after insert
on regular for each row
begin
declare cid_t int;
declare rid_t int;
declare type_t varchar (20);
declare customer_t int;
```

```
declare date_paid_t date;
select type into type_t
from delivery where tid= new.tid;
select cid into cid_t
from delivery where tid = new.tid;
select rid into rid_t
from delivery where tid = new.tid;
select is_company into customer_t
from customer where cid = cid_t;
select initiated_date into date_paid_t
from delivery where tid = new.tid;
if customer_t = 0 then
insert into bill
values(new.tid, cid_t, rid_t, type_t, new.cost, "yes", date_paid_t);
elseif customer_t = 1 then
insert into bill
values(new.tid, cid_t, rid_t, type_t, new.cost, "no", NULL);
end if;
end $$
delimiter ;
```

/*Trigger on after insertion on international to update the bill value. All details like tid, cid, rid are procured from different relevant tables. The promised date and initiated date are calculated and inserted in the BILL table along with the cost calculated. Pay status is set as “yes” (meaning paid) to normal customers and “no” (meaning not paid) to companies which have a pact with the delivery agency (and hence pay monthly)*/

```
delimiter $$
create trigger after_hazardous_insert
after insert
on hazardous for each row
begin
declare cid_t int;
declare rid_t int;
declare type_t varchar (20);
declare customer_t int;
declare date_paid_t date;
select type into type_t
from delivery where tid= new.tid;
select cid into cid_t
from delivery where tid = new.tid;
select rid into rid_t
from delivery where tid = new.tid;
select is_company into customer_t
from customer where cid = cid_t;
select initiated_date into date_paid_t
from delivery where tid = new.tid;
if customer_t = 0 then
```

```
insert into bill
values(new.tid, cid_t, rid_t, type_t, new.cost, "yes", date_paid_t);
elseif customer_t = 1 then
insert into bill
values(new.tid, cid_t, rid_t, type_t, new.cost, "no", NULL);
end if;
end $$
delimiter ;
/*Loading data into the delivery table*/
LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\delivery_data.csv' INTO TABLE delivery FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n';

/*Trigger after insert in delivery status to insert into delivered if the satus of any entry for a given tid and given date is “Delivered”*/
delimiter $$
create trigger after_delivery_status_insert
after insert
on delivery_status for each row
begin
declare date_prm date;
declare date_deli date;
declare type_t varchar (15);
select date_t into date_deli
from delivery_status where tid = new.tid and status = new.status;
select type into type_t
from delivery where tid = new.tid;
```

```
if type_t = "international" then
select promised_date into date_prm
from international where tid = new.tid;
elseif type_t = "regular" then
select promised_date into date_prm
from regular where tid = new.tid;
elseif type_t = "hazardous" then
select promised_date into date_prm
from hazardous where tid = new.tid;
end if;
if new.status = "Delivered" then
insert into delivered
values (new.tid, date_prm, date_deli, new.transport_number, new.location);
end if;
end $$
delimiter ;

/*Trigger on after delivered insert. The num_deliveries column of the customer
and the recepients are incremented*/
delimiter $$
create trigger after_delivered_insert
after insert
on delivered for each row
begin
declare cid_t int;
declare rid_t int;
declare nd_c int;
```

```
declare nd_r int;
select cid into cid_t
from delivery where tid = new.tid;
select rid into rid_t
from delivery where tid = new.tid;
select num_deliveries into nd_c
from customer where cid = cid_t;
set nd_c = nd_c + 1;
update customer
set num_deliveries = nd_c
where cid = cid_t;
select num_deliveries into nd_r
from recepient where rid = rid_t;
set nd_r = nd_r + 1;
update recepient
set num_deliveries = nd_r
where rid = rid_t;
end $$
delimiter ;
```

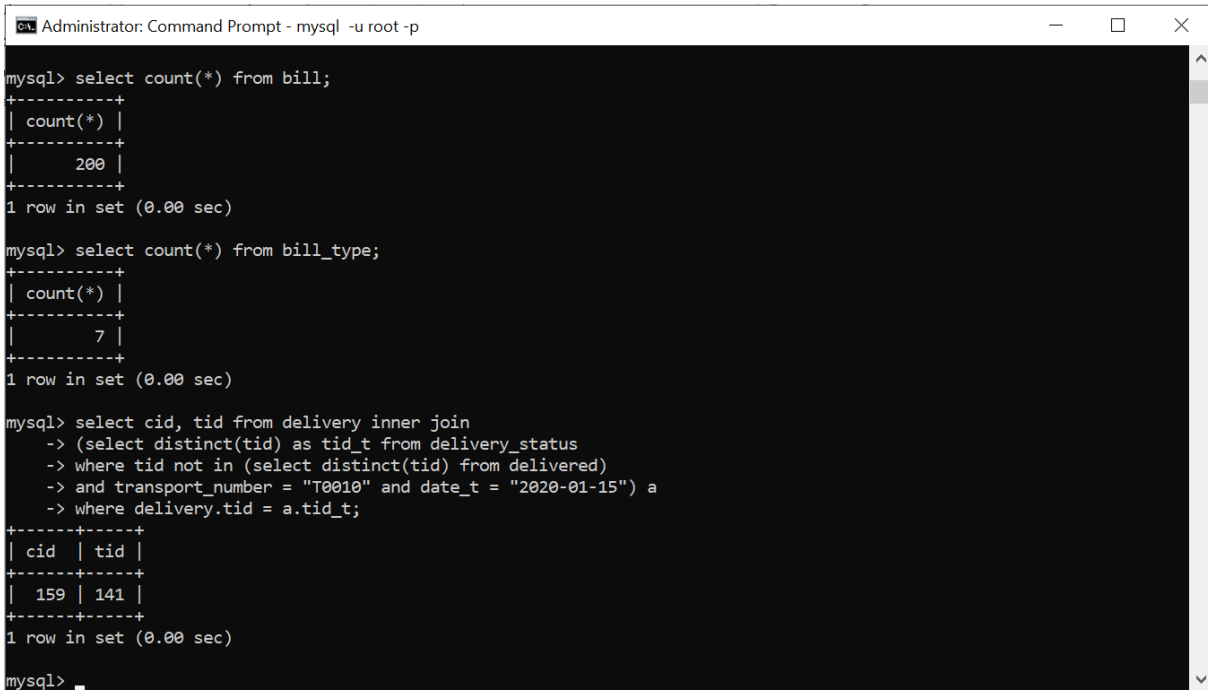
QUERIES

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.

```
select cid, tid from delivery inner join
(select distinct(tid) as tid_t from delivery_status
where tid not in (select distinct(tid) from delivered)
and transport_number = "T0010" and date_t = "2020-01-15") a
```

where delivery.tid = a.tid_t;




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

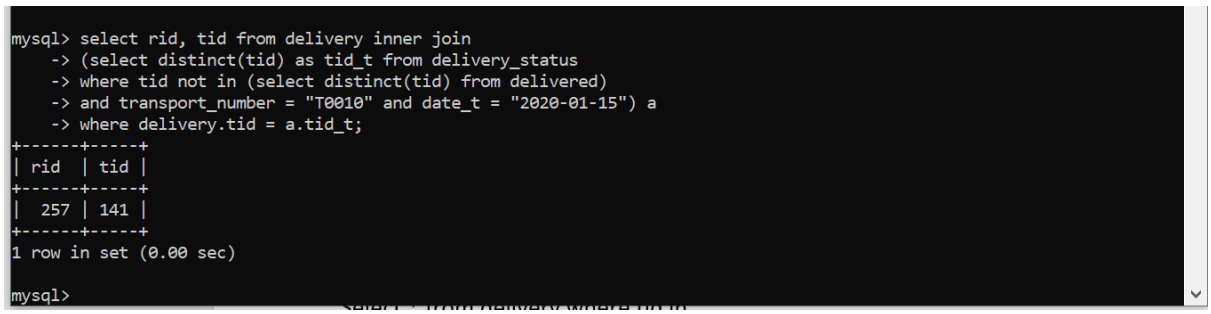
select rid, tid from delivery inner join

(select distinct(tid) as tid_t from delivery_status

where tid not in (select distinct(tid) from delivered)

and transport_number = "T0010" and date_t = "2020-01-15") a

where delivery.tid = a.tid_t;




1) c) Find the last successful delivery by that truck prior to the crash.

Select * from delivery where tid in

(select tid from delivered where transport_number = "T0010" and delivered_date = "2020-04-23");

```
mysql> Select * from delivery where tid in
    -> (select tid from delivered where transport_number = "T0010" and delivered_date = "2020-04-23");
+-----+------+------+---------------+----------------+
| tid | cid  | rid  | type          | initiated_date |
+-----+------+------+---------------+----------------+
|  63 |  228 |  265 | international | 2020-04-21     |
| 175 |  154 |  213 | hazardous     | 2020-04-22     |
+-----+------+------+---------------+----------------+
2 rows in set (0.00 sec)

mysql>
```

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

delimiter $$

create procedure custMostShip()

begin

declare cid_t int;

declare max_count int;

select max(b.count_t) into max_count from

(select a.cid, count(a.cid) as count_t from

(select cid from bill where datediff(sysdate(), paid_date) < 365) a

group by(a.cid)) b;

select cid, cname from customer where cid in

(select a.cid from

(select cid from bill where datediff(sysdate(), paid_date) < 365) a

group by(a.cid)

having count(a.cid) = max_count);

end $$

delimiter ;

call custMostShip();

```
mysql> call custMostShip();
+-----+------------------+
| cid | cname            |
+-----+------------------+
|   9 | Jennifer Williams |
|  71 | Antonio Howard    |
|  81 | Allen Larson      |
| 260 | Dr. Connie Wilson |
+-----+------------------+
4 rows in set (0.01 sec)

Query OK, 0 rows affected (0.02 sec)

mysql>
```

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

```
delimiter $$

create procedure custMostMoney()

begin

declare cid_t int;

declare max_sum int;

select max(b.sum_t) into max_sum from

(select a.cid, sum(a.amount_due) as sum_t from

(select cid, amount_due from bill where datediff(sysdate(), paid_date) < 365) a

group by(a.cid)) b;

select a.cid into cid_t from

(select cid, amount_due from bill where datediff(sysdate(), paid_date) < 365) a

group by(a.cid)

having sum(a.amount_due) = max_sum;

select cid, cname from customer

where cid = cid_t;

end $$

delimiter ;


call custMostMoney();
```

```
mysql> call custMostMoney();
+-----+------------------+
| cid | cname            |
+-----+------------------+
| 260 | Dr. Connie Wilson |
+-----+------------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)

mysql>
```

4) Find the street with the most customers.

```
select street_address from customer

group by (street_address)

having count(cid) =

(select max(a.count_street) as max_count from

(select cid, street_address, count(*) as count_street from customer

group by (street_address)) a);
```

```
mysql> select street_address from customer
    -> group by (street_address)
    -> having count(cid) =
    -> (select max(a.count_street) as max_count from
    -> (select cid, street_address, count(*) as count_street from customer
    -> group by (street_address)) a);
+------------------------------+
| street_address               |
+------------------------------+
| 25564 Cook Via Suite 262 Lake |
+------------------------------+
1 row in set (0.00 sec)

mysql>
```

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

```
select tid, delivered_date, promised_date from delivered

where datediff(delivered_date, promised_date) > 0;
```

```
mysql> select tid, delivered_date, promised_date from delivered
    -> where datediff(delivered_date, promised_date) > 0;
+-----+----------------+---------------+
| tid | delivered_date | promised_date |
+-----+----------------+---------------+
| 180 | 2020-10-27     | 2020-10-26    |
+-----+----------------+---------------+
1 row in set (0.00 sec)

mysql>
```

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

```
delimiter $$

create procedure custDetails(in cid_t int)

begin

select customer.cid, cname, street_address, city, phone_number,
sum(amount_due), pay_status

from customer, bill

where customer.cid = bill.cid

and customer.cid = cid_t

group by (bill.cid);

end $$

delimiter ;
```

```
call custDetails(101);
```

```
mysql> call custDetails(101);
+-----+--------------+------------------------------+--------------+----------------+----------------+------------
+
| cid | cname        | street_address               | city         | phone_number   | sum(amount_due) | pay_status
|
+-----+--------------+------------------------------+--------------+----------------+----------------+------------
+
| 101 | Devin Ferrell | 7915 Mitchell Manors Suite 325 | South Kristen | 420-536-7922x38 |            600 | yes
|
+-----+--------------+------------------------------+--------------+----------------+----------------+------------
+
1 row in set (0.01 sec)

Query OK, 0 rows affected (0.01 sec)

mysql>
```

7) A bill listing charges by type of service.

```
create table bill_type(

type varchar (20),

type_in varchar (20),

cost int);
```

LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\bill_type_data.csv' INTO TABLE bill_type FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n';

Select * from bill_type;

```
mysql> Select * from bill_type;
+---------------+----------------+------+
| type          | type_in        | cost |
+---------------+----------------+------+
| international | fast           | 1500 |
| international | standard       | 1200 |
| regular       | small envelope |  200 |
| regular       | small box      |  600 |
| regular       | large box      |  800 |
| hazardous     | chemicals      | 1000 |
| hazardous     | harmless       |  800 |
+---------------+----------------+------+
7 rows in set (0.00 sec)

mysql>
```

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

select * from bill;

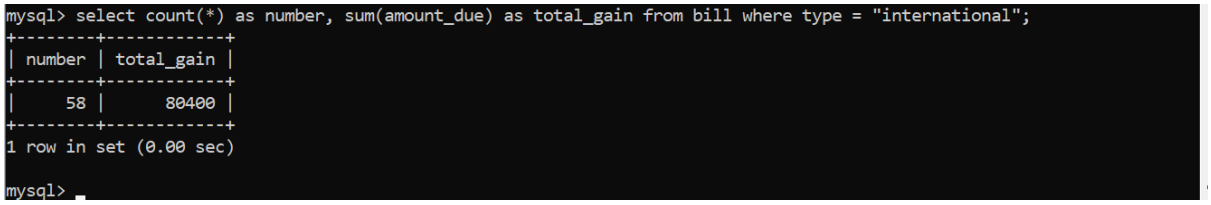
```
mysql> select * from bill;
+-----+------+------+---------------+------------+------------+------------+
| tid | cid  | rid  | type          | amount_due | pay_status | paid_date  |
+-----+------+------+---------------+------------+------------+------------+
|   1 |  385 |  238 | international |       1500 | no         | NULL       |
|   2 |  228 |   76 | hazardous     |       1000 | yes        | 2020-06-24 |
|   3 |   91 |    6 | hazardous     |       1000 | yes        | 2020-09-21 |
|   4 |   40 |  227 | international |       1500 | yes        | 2020-10-26 |
|   5 |  265 |   66 | hazardous     |       1000 | yes        | 2020-07-19 |
|   6 |  259 |  170 | regular       |        600 | yes        | 2020-05-18 |
|   7 |  172 |   11 | international |       1200 | yes        | 2020-06-06 |
|   8 |   83 |  238 | hazardous     |        800 | yes        | 2020-05-15 |
|   9 |  376 |   87 | regular       |        600 | no         | NULL       |
|  10 |  381 |  113 | hazardous     |       1000 | no         | NULL       |
|  11 |  274 |   93 | international |       1500 | yes        | 2020-08-20 |
|  12 |   81 |  243 | regular       |        600 | yes        | 2020-03-17 |
|  13 |  219 |  161 | hazardous     |       1000 | yes        | 2020-09-04 |
|  14 |  357 |  149 | hazardous     |        800 | no         | NULL       |
|  15 |  170 |  195 | regular       |        800 | yes        | 2020-03-06 |
|  16 |   61 |  175 | regular       |        200 | yes        | 2020-06-24 |
|  17 |   11 |  251 | international |       1200 | yes        | 2020-03-10 |
|  18 |   63 |  114 | international |       1200 | yes        | 2020-02-05 |
|  19 |  316 |  120 | hazardous     |        800 | no         | NULL       |
|  20 |  104 |  149 | regular       |        200 | yes        | 2020-03-31 |
|  21 |  246 |  124 | hazardous     |        800 | yes        | 2020-08-01 |
|  22 |  145 |  274 | hazardous     |        800 | yes        | 2020-10-19 |
|  23 |  244 |  278 | hazardous     |       1000 | yes        | 2020-10-22 |
|  24 |  354 |   24 | international |       1500 | no         | NULL       |
|  25 |  320 |   15 | hazardous     |       1000 | no         | NULL       |
```

ADDITIONAL QUERIES

1) Summary of total number of revenue collected by type of the service (international, hazardous, regular)

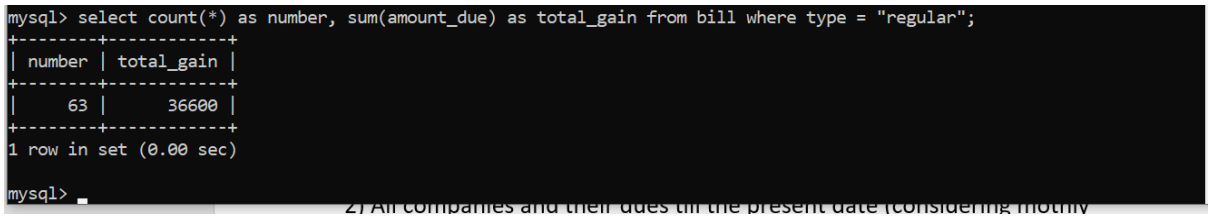
a) international

select count(*) as number, sum(amount_due) as total_gain from bill where type = "international";



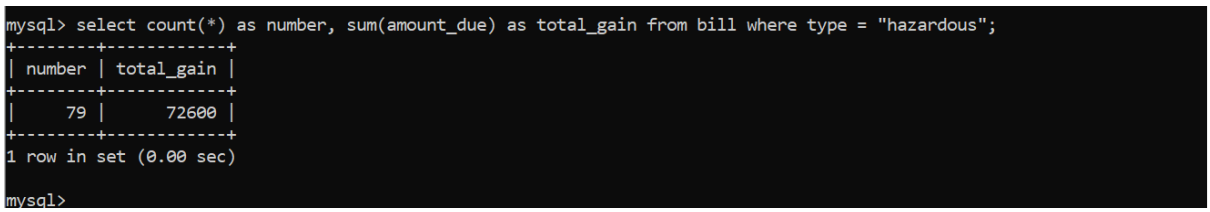

b) regular

select count(*) as number, sum(amount_due) as total_gain from bill where type = "regular";




c) hazardous

select count(*) as number, sum(amount_due) as total_gain from bill where type = "hazardous";




2) All companies and their dues till the present date (considering mothly payment)

select cid, sum(amount_due) from bill where tid in

(select tid from delivery

where cid in (select cid from customer where is_company = 1)

and datediff(sysdate(), initiated_date) > 30)

group by (cid);

```
Administrator: Command Prompt - mysql  -u root -p
mysql> select cid, sum(amount_due) from bill where tid in
    -> (select tid from delivery
    -> where cid in (select cid from customer where is_company = 1)
    -> and datediff(sysdate(), initiated_date) > 30)
    -> group by (cid);
+------+----------------+
| cid  | sum(amount_due) |
+------+----------------+
|  303 |            800 |
|  308 |            800 |
|  311 |           1500 |
|  312 |           1500 |
|  314 |            800 |
|  316 |            800 |
|  317 |            600 |
|  319 |           2000 |
|  320 |           1000 |
|  325 |            800 |
|  333 |            800 |
|  334 |            800 |
|  335 |            400 |
|  339 |            800 |
|  340 |            800 |
|  341 |            800 |
|  344 |            800 |
|  348 |           1000 |
|  349 |           1000 |
|  354 |           1500 |
|  357 |           1600 |
|  358 |           1200 |
|  361 |           1200 |
|  364 |           1500 |
|  366 |            200 |
|  372 |            800 |
|  373 |           1200 |
|  374 |            800 |
|  376 |            600 |
|  378 |           1200 |
|  379 |            800 |
|  381 |           1000 |
|  385 |           1500 |
```

3) Customer with most number of deliveries till now

select cid, cname, num_deliveries from customer where num_deliveries in (select max(num_deliveries) from customer);

```
mysql> select cid, cname, num_deliveries from customer where num_deliveries in (select max(num_deliveries) from customer
);
+-----+------------------+----------------+
| cid | cname            | num_deliveries |
+-----+------------------+----------------+
|   2 | Maria Jackson    |              2 |
|   9 | Jennifer Williams |              2 |
|  81 | Allen Larson     |              2 |
| 129 | Carlos Peters    |              2 |
| 245 | Karen Taylor     |              2 |
+-----+------------------+----------------+
5 rows in set (0.00 sec)

mysql>
```

4) Recepients who have received the most deliveries till now

select rid, rname, num_deliveries from recepient where num_deliveries in (select max(num_deliveries) from recepient);

```
mysql> select rid, rname, num_deliveries from recepient where num_deliveries in (select max(num_deliveries) from recepie
nt);
+-----+-----------+----------------+
| rid | rname     | num_deliveries |
+-----+-----------+----------------+
|   1 | Andrea Lee |              3 |
+-----+-----------+----------------+
1 row in set (0.00 sec)

mysql>
```