# DBMS LAB PROJECT

(CACS05)

# Water Refill Management System

**PROJECT BY**

- **Sahil khan 2022UCA1954**

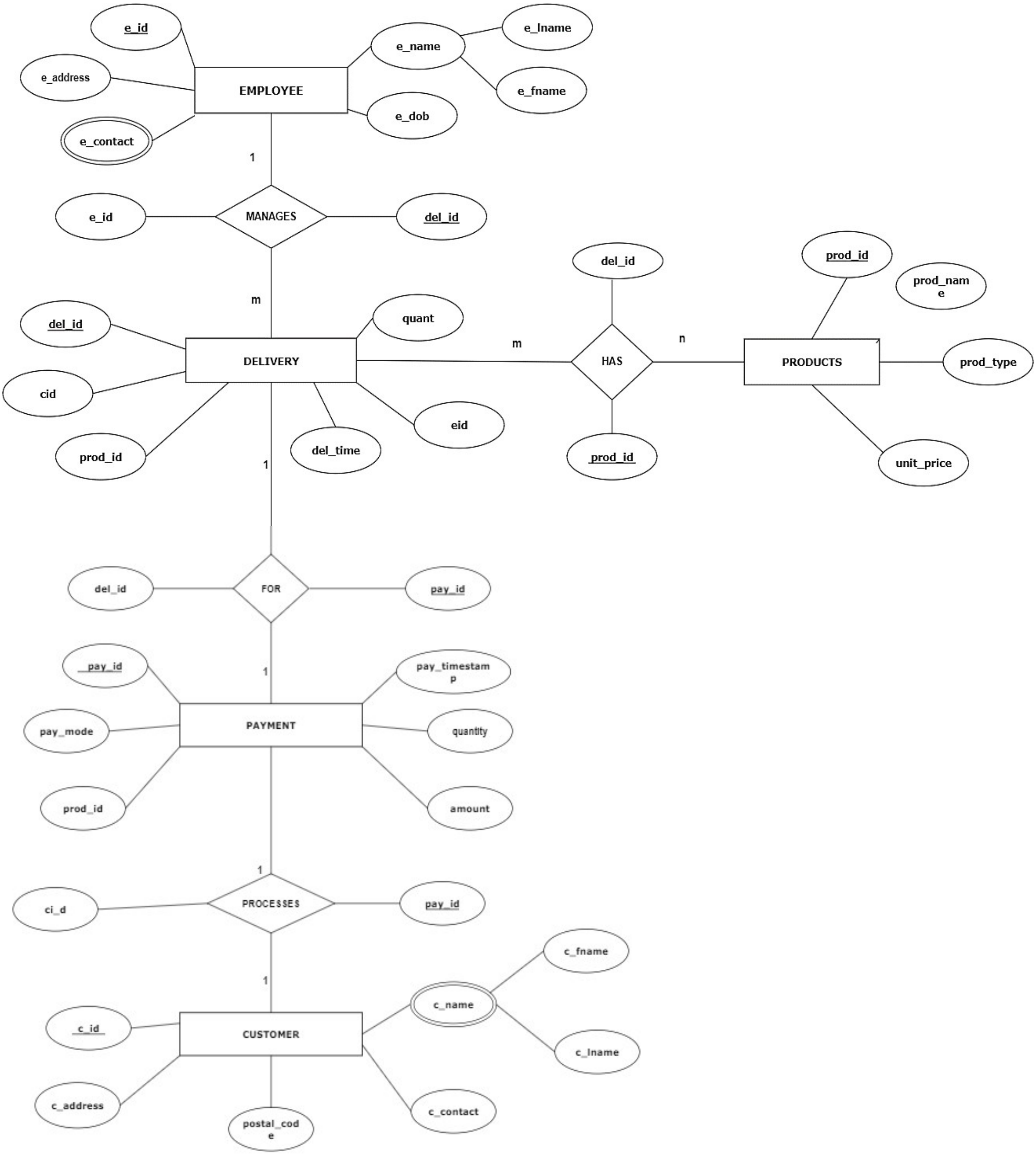
# Problem Statement

To solve this problem there comes one of the fastest and expanding businesses which is water refilling station. Water refilling station is small water system that has its own water purification facility producing a portable drinking water. The aqua water refilling system has their own water tank and equipment that intend on their business So, we design a system in this kind of business in order to be on top and align on the fast-growing business that is demand now a day. The purpose of water refilling management system is to overcome difficulties in manual operation in refilling station. This system manages to display the data to be filled by the user according to the information of the customer in organize manner, such that their personal details, and the services they want to avail as well as the payment on the transaction they purchased. The system keeps the information of the customer and the details of what they purchase. The system coordinates the arrangement on delivery of products. It consistsall the records for the location of the clients, date of transaction, schedule of delivery, contact number andthe person assign to deliver and the payment of customer to the quantity of product that about to deliver. The system also views the information about the availability of the pro ducts as well as the containers. The system views the available containers to provide stocks again. This system also manages the information ofthe employees that a refilling station must have just like front liner, cashier, technical assistant, and delivery. Upon having this system, it will provide the capacity to the owner and clients to transact without spending time and effort.

**Basic features:**

- **Manage Employee:** which is part of enterprise and has attributes id which can be uniquely used to identify him/her, name, date of birth , age(automatically obtained from date of birth), address, contact (may be multivalued).
- **Manage Customer:** which has customer id to identify him/her, name, address, contact(may be multivalued), date of birth, age(automatically obtained from date of birth)
- **Manage product:** product id used to identify uniquely the product, its name, its type, unitprice .
- **Manage Payment:** which has pay_id to uniquely identify tuple, product id to identifywhich product was purchased, quantity of product purchased, pay_mode, amount, payment(date and time).
- **Manage Delivery:** which has delievery_id to identify unique tuple, product id (which product was delivered), customer id to whom it was delivered related to customer table, quantity , delivery time, employee id(who delivered the bottle).

# ER Diagram for this problem statement



# Converting this ER diagram to Relational Model

- **employee** (eid, efname, elname, eaddress, e_dob, e_contact)
- **product** (prod_id , prod_type, prod_name, unit_price)
- **customer** (cid, cfname, clname,  caddress, postal_code, c_contact)
- **delivery** (del_id , cid, prod_id, eid, quant, del_time)
- **payment** (pay_id, pay_mode, prod_id, pay_timestamp, quantity, amount)
- **manages** ( eid, del_id)
- **processes** (cid, pay_id)
- **for** (del_id, pay_id)
- **has** (del_id, prod_id)

## Reducing tables by Mapping Cardinalities

- In our ER diagram employee manages delivery is 1:m relationship therefore we can combine delivery with relationship manages with del_id as primary key.
- In our ER diagram customer processes payment is 1:1 relationship therefore we can combine payment with processes with pay_id as primary key.
- In our ER diagram payment for delivery is 1:1 relationship therefore we cancombine payment with for with pay_id as primary key

### Now Updated Relational Schema

- **employee** (eid, efname, elname, eaddress, e_dob, e_contact)
- **product** (prod_id , prod_type, prod_name, unit_price)
- **customer** (cid, cfname, clname,  caddress, postal_code, c_contact)
- **delivery** (del_id , cid, prod_id, eid, quant, del_time,eid)

- **payment** (pay_id, pay_mode, cid, prod_id, pay_timestamp, quantity, del_id, amount)
- **has** (del_id, prod_id)

## Listing all the Functional Dependencies

- **employee** (eid, efname, elname, eaddress, e_dob, e_contact)

In this table eid is primary key so it determines all other columnsNo other functional dependency is there

- **product** (prod_id , prod_type, prod_name, unit_price)

In this table prod_id is primary key so it determine all other attributes. No other functional dependency is there

- **customer** (cid, cfname, clname,  caddress, postal_code, c_contact)

In this table cid is primary key so it determines all the other attributes. Also caddress can determine postal_code. So there are 2 FDs in total

- **delivery** (del_id , cid, prod_id, eid, quant, del_time,eid)

In this table del_id is FD so it determines all other attributes except this there is no FD

- **payment** (pay_id, pay_mode, cid, prod_id, pay_timestamp, quantity, del_id, amount)

In this table pay_id is primary key so it determines all other attributes. Also prod_id and quantity collectively determine amount. So in total there are 2 dependencies in this relation.

- **has** (del_id, prod_id)

In this both del_id and prod_id are collectively primary key so there is no FD

Now after seeing all relations final functional dependecy list is –

- eid -> efname, elname, eaddress, e_dob, e_contact
- prod_id -> prod_type, prod_name, unit_price
- cid -> cfname, clname, caddress, postal_code, c_contact
- caddress -> postal_code
- del_id -> cid, prod_id, eid, quant, del_time,eid
- pay_id -> , pay_mode, cid, prod_id, pay_timestamp, quantity, del_id, amount
- ( prod_id quant) -> amount

## Conversion To First Normal Form

A table is in 1st Normal Form if:

- There are only Single Valued Attributes.
- Attribute Domain does not change.
- There is unique name for every Attribute/Column.
- The order in which data is stored does not matter

In our database only e-contact attribute of Employee table is a multi-valued attribute therefore we make a separate table to represent multi-valued attribute containing attributes e_id, e_contact to normalize it in 1stNormal Form.

So Tables with their attributes are as follows :

- **employee** (eid, efname, elname, eaddress, e_dob)
- **employee_contact** (eid, e_contact)
- **product** (prod_id , prod_type, prod_name, unit_price)
- **customer** (cid, cfname, clname, caddress, postal_code, c_contact)

- **delivery** (del_id , cid, prod_id, eid, quant, del_time,eid)
- **payment** (pay_id, pay_mode, cid, prod_id, pay_timestamp, quantity, del_id, amount)
- **has** (del_id, prod_id)

## Conversion To Second Normal Form

A database is in second normal form if it satisfies the following conditions:

- It is in first normal form
- All non-key attributes are fully functional dependent on the candidate key

In a table, if attribute B is functionally dependent on A, but is not functionallydependent on a proper subset of A, then B is considered fully functional dependent on A.

We have now converted our database into the First Normal Form already and as all functional dependencies are fully functional dependent on the candidate key because each of our candidate key is a single key attribute and it is not composite therefore no modifications are required and our database is already in second normal form.

## Conversion To Third Normal Form

A table is in 3NF, only if a relation is in 2NF and it has no Transitive Functional Dependency This condition will be satisfied if Left Hand Side of Functional Dependency there will be a Candidate Key or Primary Key **OR** Right Hand Side ofthe Functional Dependency is a Primary Key attribute.

In our database we have transitive dependency in Customer and Payment Table.

**Resolving Transitive Dependency**

1. **Customer Table**

Here cid determines address and address determines postal_code. We resolvethis problem by first assuming that e_contact is unique and there can be manyaddresses for one econtact. The table is divided into 2 sub- tables:

- **customer** ( cid, cfname, clname, c_contact, caddress )Primary Key: cid
  Foreign Key: caddress

- **customer_location** ( address, postal_code )Primary Key: address
  postal_code

**2. Payment Table**

Here pay_id determines quant and prod_id. quant and prod_id together determines amount. We resolve the problem by decomposing table into 2 tables:

- **payment_details(** pay_id, pay_mode, cid, prod_id, quant, pay_time, del_id)Primary Key: pay_id
  Foreign Key:  quant prodid

- **pay_amount** ( quant, prod_id , amount)Primary Key: quant prod_id

## Conversion to BCNF(Boyce-Codd Normal Form)

CNF(Boyce - Codd Normal Form) in DBMS is an advanced version of **3NF** (third normal form). A table or a relation is said to be in BCNF in DBMS if the table or the relation is already in 3NF, and also, for every functional dependency (say, X->Y), X is either the super key or the candidate key. In simple terms, for any case (say, X->Y), X can't be a non-prime attribute.

The given database with table as well as attributes is as follows :

- **Employee** ( eid, efname, elname, eaddress, dob)Prime Attributes: eid
  Non- Prime Attributes: enflame, enlname, eaddress, dob

- **Employee_contact** ( eid, contact )Prime Attributes: eid

Non - Prime Attributes: contact

- **product** (prod_id , prod_type, prod_name, unit_price)Prime Attributes: prod_id
  Non - Prime Attributes: prod_name,prod_name,unit_Price

- **customer** ( cid, cfname, clname, c_contact, caddress )Prime Attributes: cid
  Non - Prime Attributes: cfname, clname, c_contact, caddress
- **customer_location** ( address, postal_code )Prime Attributes: address
  postal_code Non - Prime Attributes: N/A

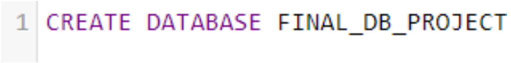
- **delivery** (del_id , cid, prod_id, eid, quant, del_time,eid)Prime Attributes:  del_id
  None - Prime Attributes: cid, eid, prod_id, quant, del_time

- **payment_details(** pay_id, pay_mode, cid, prod_id, quant, pay_time, del_id)Prime Attributes: pay_id
  None - Prime Attributes: pay_mode, cid, prod_id,quant, pay_time, del_id

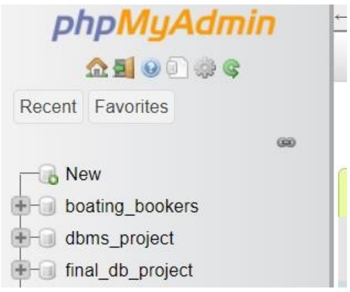
- **pay_amount** ( quant, prod_id , amount)Prime Attributes:  quant,
  prod_id Non - Prime Attributes:
  amount

As no Non-Primary Attribute is determining any Prime Attribute in any giventable the database is in BCNF Form already.

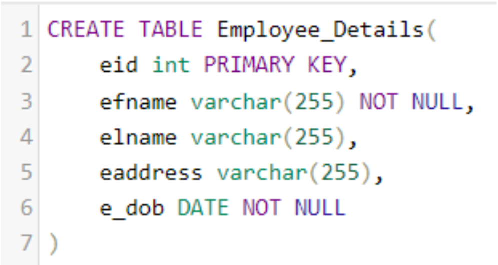
## Creating Database –



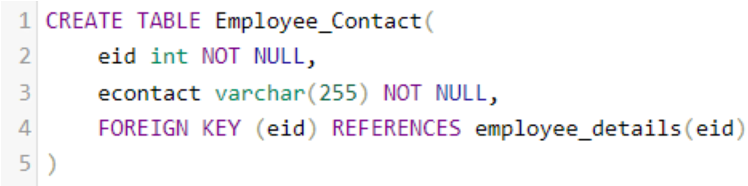
```
CREATE DATABASE FINAL_DB_PROJECT
```



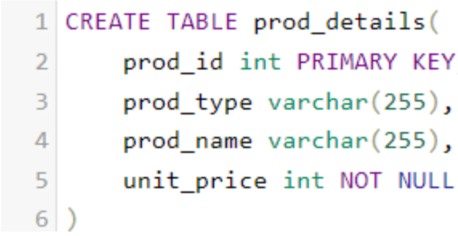
## Creating Table –



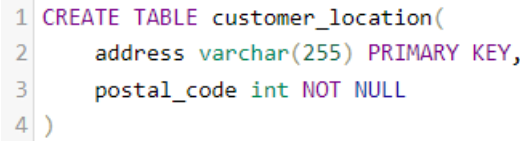
```
CREATE TABLE Employee_Details(
    eid int PRIMARY KEY,
    efname varchar(255) NOT NULL,
    elname varchar(255),
    eaddress varchar(255),
    e_dob DATE NOT NULL
)
```



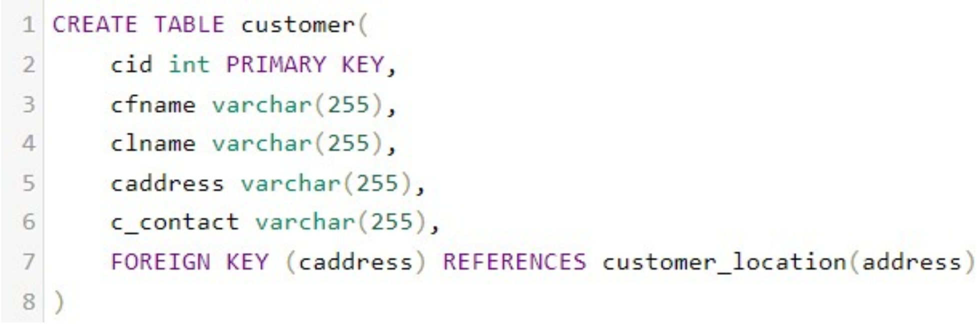
```
CREATE TABLE Employee_Contact(
    eid int NOT NULL,
    econtact varchar(255) NOT NULL,
    FOREIGN KEY (eid) REFERENCES employee_details(eid)
)
```



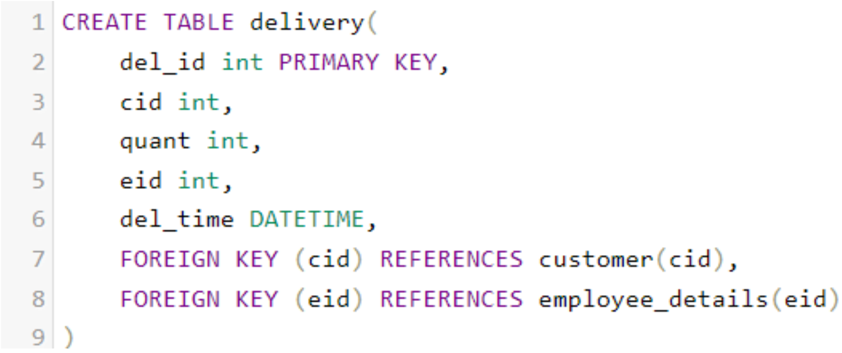
```
CREATE TABLE prod_details(
    prod_id int PRIMARY KEY,
    prod_type varchar(255),
    prod_name varchar(255),
    unit_price int NOT NULL
)
```



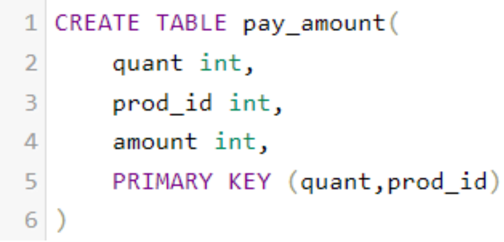
```
CREATE TABLE customer_location(
    address varchar(255) PRIMARY KEY,
    postal_code int NOT NULL
)
```



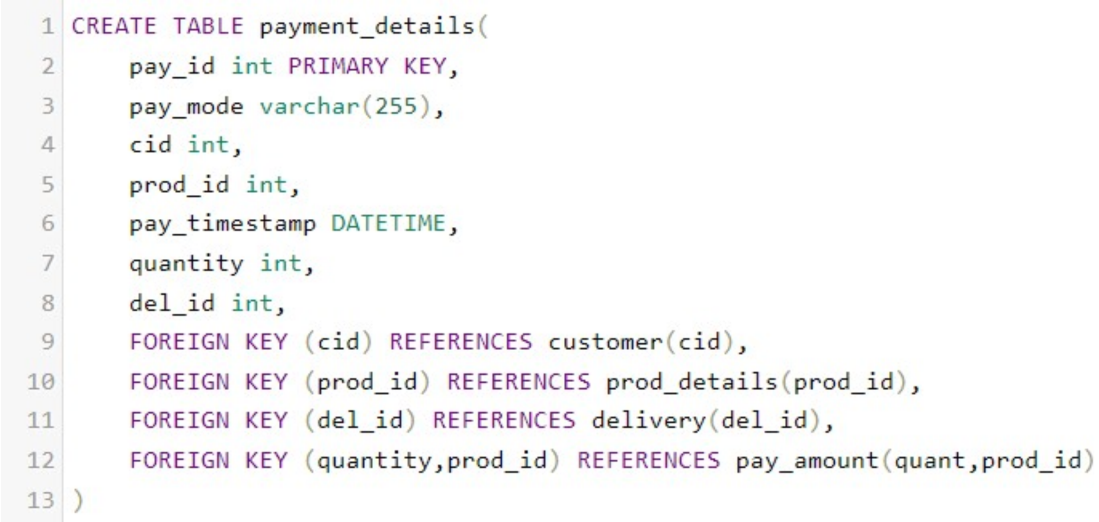
```
CREATE TABLE customer(
    cid int PRIMARY KEY,
    cfname varchar(255),
    clname varchar(255),
    caddress varchar(255),
    c_contact varchar(255),
    FOREIGN KEY (caddress) REFERENCES customer_location(address)
)
```



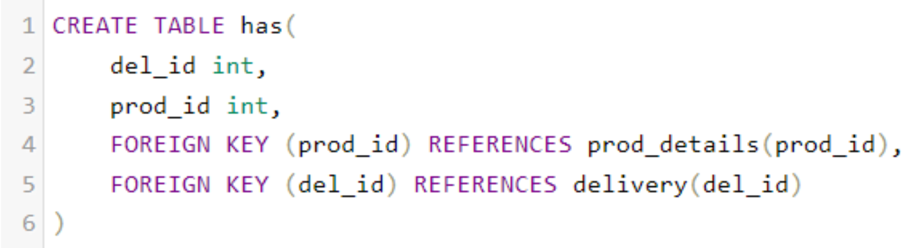
```
CREATE TABLE delivery(
    del_id int PRIMARY KEY,
    cid int,
    quant int,
    eid int,
    del_time DATETIME,
    FOREIGN KEY (cid) REFERENCES customer(cid),
    FOREIGN KEY (eid) REFERENCES employee_details(eid)
)
```



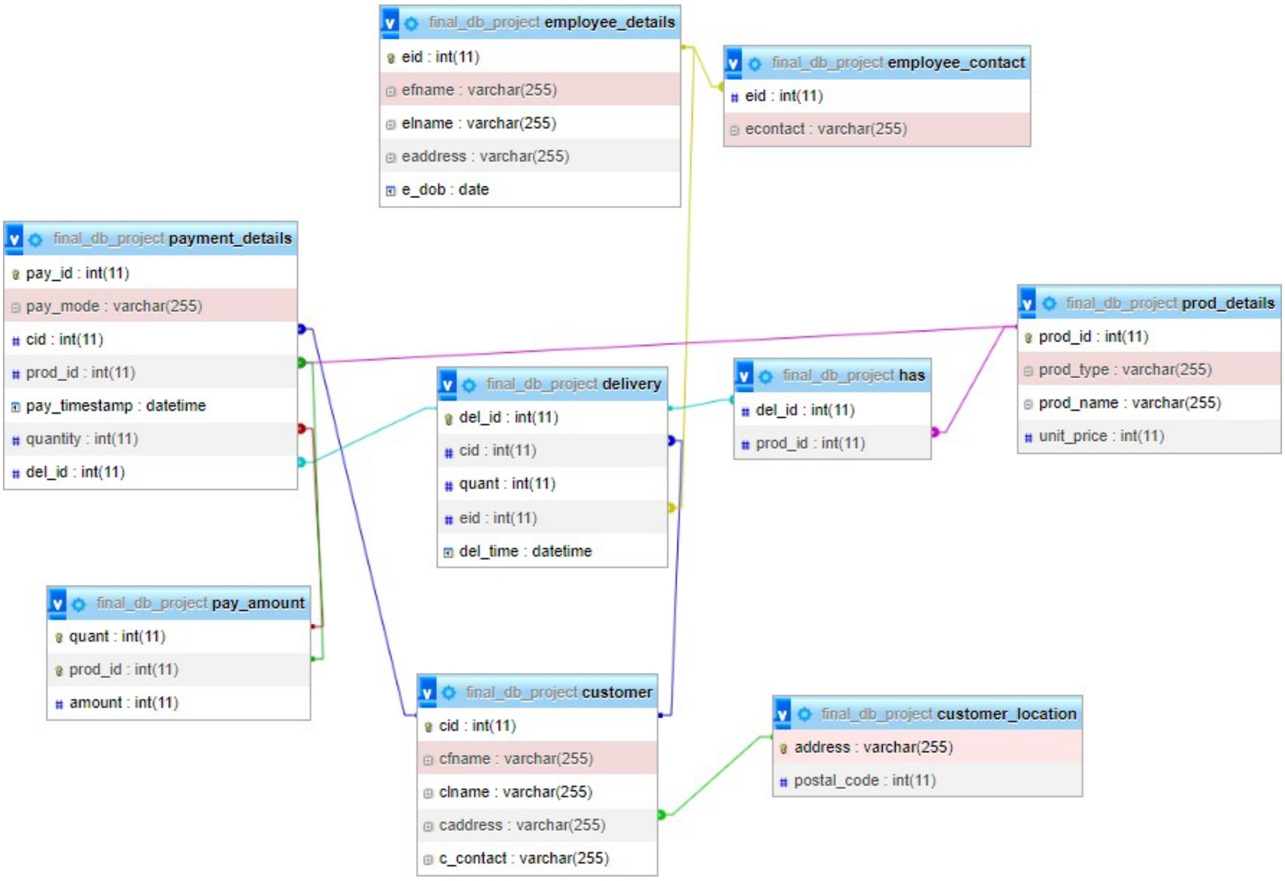
```sql
CREATE TABLE pay_amount(
    quant int,
    prod_id int,
    amount int,
    PRIMARY KEY (quant,prod_id)
)
```



```sql
CREATE TABLE payment_details(
    pay_id int PRIMARY KEY,
    pay_mode varchar(255),
    cid int,
    prod_id int,
    pay_timestamp DATETIME,
    quantity int,
    del_id int,
    FOREIGN KEY (cid) REFERENCES customer(cid),
    FOREIGN KEY (prod_id) REFERENCES prod_details(prod_id),
    FOREIGN KEY (del_id) REFERENCES delivery(del_id),
    FOREIGN KEY (quantity,prod_id) REFERENCES pay_amount(quant,prod_id)
)
```



```sql
CREATE TABLE has(
    del_id int,
    prod_id int,
    FOREIGN KEY (prod_id) REFERENCES prod_details(prod_id),
    FOREIGN KEY (del_id) REFERENCES delivery(del_id)
)
```



final_db_project employee_details
eid : int(11)
efname : varchar(255)
elname : varchar(255)
eaddress : varchar(255)
e_dob : date

final_db_project employee_contact
eid : int(11)
econtact : varchar(255)

final_db_project payment_details
pay_id : int(11)
pay_mode : varchar(255)
cid : int(11)
prod_id : int(11)
pay_timestamp : datetime
quantity : int(11)
del_id : int(11)

final_db_project prod_details
prod_id : int(11)
prod_type : varchar(255)
prod_name : varchar(255)
unit_price : int(11)

final_db_project delivery
del_id : int(11)
cid : int(11)
quant : int(11)
eid : int(11)
del_time : datetime

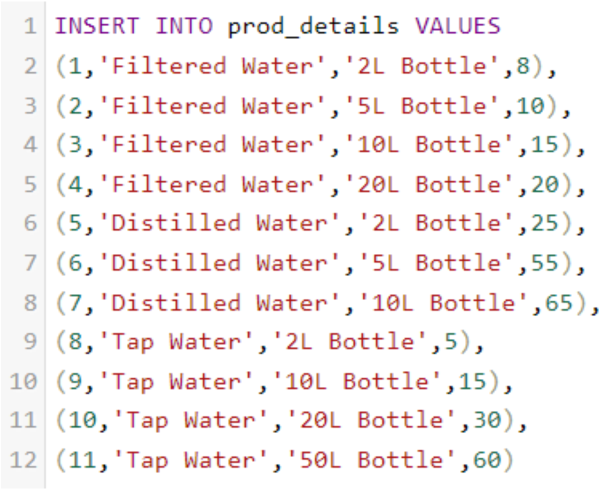
final_db_project has
del_id : int(11)
prod_id : int(11)

final_db_project pay_amount
quant : int(11)
prod_id : int(11)
amount : int(11)

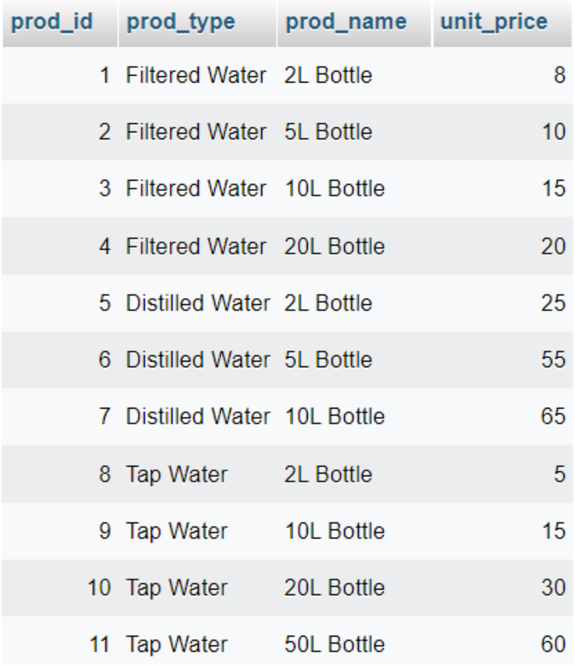
final_db_project customer
cid : int(11)
cfname : varchar(255)
clname : varchar(255)
caddress : varchar(255)
c_contact : varchar(255)

final_db_project customer_location
address : varchar(255)
postal_code : int(11)

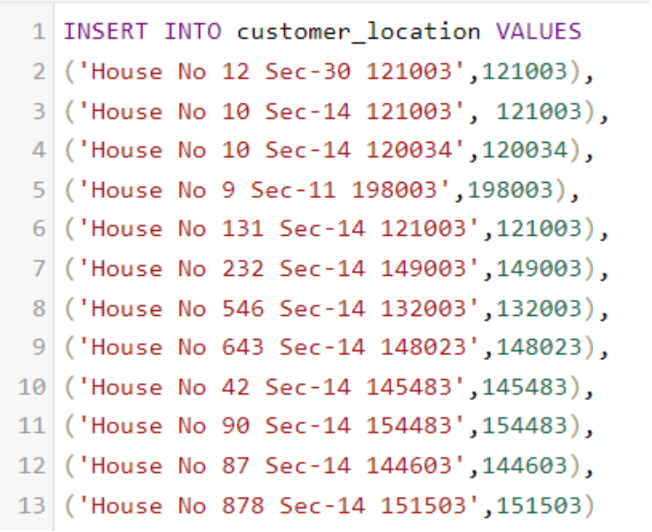
# *Inserting Values in Database Tables*



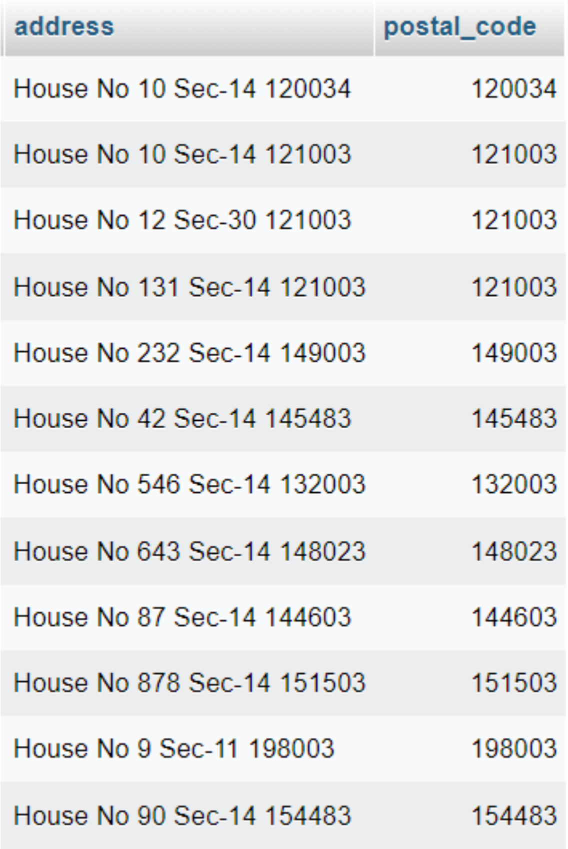
```sql
INSERT INTO prod_details VALUES
(1,'Filtered Water','2L Bottle',8),
(2,'Filtered Water','5L Bottle',10),
(3,'Filtered Water','10L Bottle',15),
(4,'Filtered Water','20L Bottle',20),
(5,'Distilled Water','2L Bottle',25),
(6,'Distilled Water','5L Bottle',55),
(7,'Distilled Water','10L Bottle',65),
(8,'Tap Water','2L Bottle',5),
(9,'Tap Water','10L Bottle',15),
(10,'Tap Water','20L Bottle',30),
(11,'Tap Water','50L Bottle',60)
```



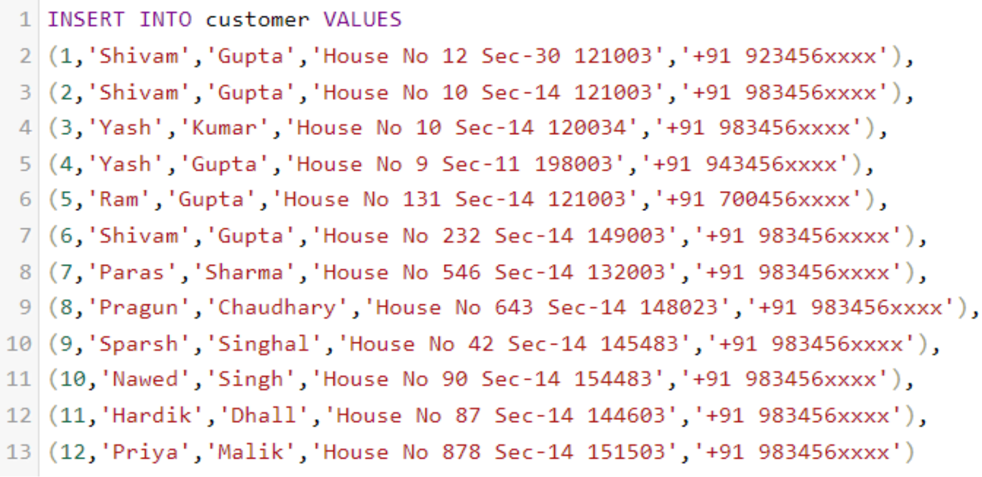
| prod_id | prod_type | prod_name | unit_price |
|---|---|---|---|
| 1 | Filtered Water | 2L Bottle | 8 |
| 2 | Filtered Water | 5L Bottle | 10 |
| 3 | Filtered Water | 10L Bottle | 15 |
| 4 | Filtered Water | 20L Bottle | 20 |
| 5 | Distilled Water | 2L Bottle | 25 |
| 6 | Distilled Water | 5L Bottle | 55 |
| 7 | Distilled Water | 10L Bottle | 65 |
| 8 | Tap Water | 2L Bottle | 5 |
| 9 | Tap Water | 10L Bottle | 15 |
| 10 | Tap Water | 20L Bottle | 30 |
| 11 | Tap Water | 50L Bottle | 60 |



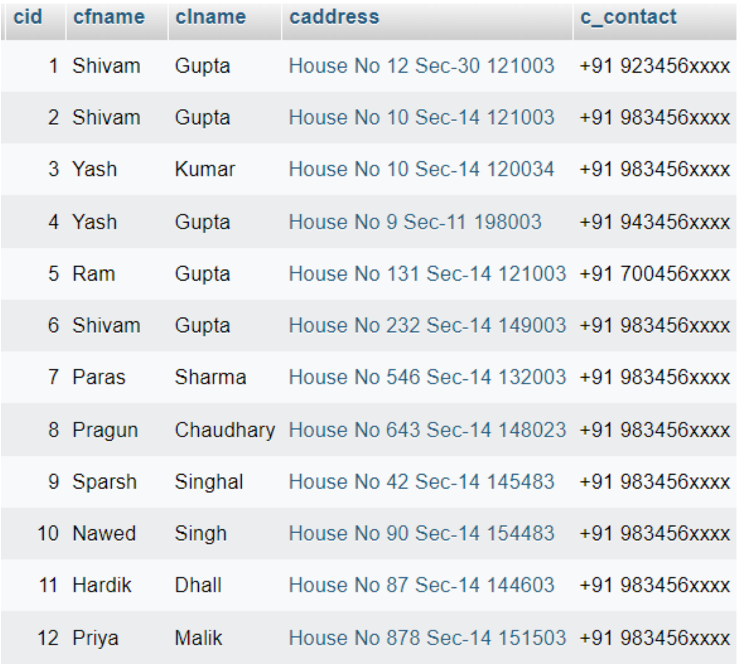
```sql
INSERT INTO customer_location VALUES
('House No 12 Sec-30 121003',121003),
('House No 10 Sec-14 121003', 121003),
('House No 10 Sec-14 120034',120034),
('House No 9 Sec-11 198003',198003),
('House No 131 Sec-14 121003',121003),
('House No 232 Sec-14 149003',149003),
('House No 546 Sec-14 132003',132003),
('House No 643 Sec-14 148023',148023),
('House No 42 Sec-14 145483',145483),
('House No 90 Sec-14 154483',154483),
('House No 87 Sec-14 144603',144603),
('House No 878 Sec-14 151503',151503)
```



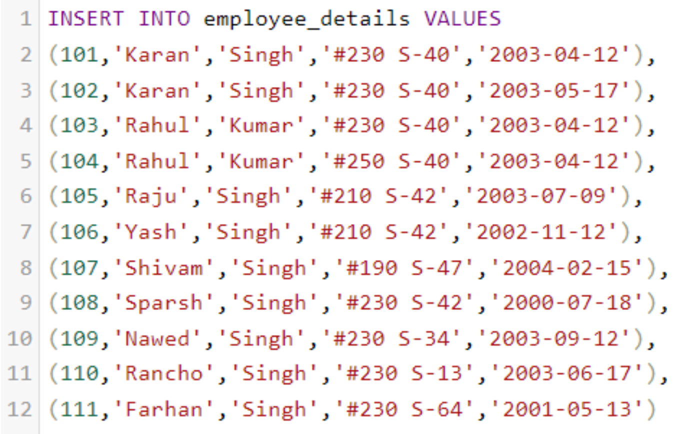
| address | postal_code |
|---|---|
| House No 10 Sec-14 120034 | 120034 |
| House No 10 Sec-14 121003 | 121003 |
| House No 12 Sec-30 121003 | 121003 |
| House No 131 Sec-14 121003 | 121003 |
| House No 232 Sec-14 149003 | 149003 |
| House No 42 Sec-14 145483 | 145483 |
| House No 546 Sec-14 132003 | 132003 |
| House No 643 Sec-14 148023 | 148023 |
| House No 87 Sec-14 144603 | 144603 |
| House No 878 Sec-14 151503 | 151503 |
| House No 9 Sec-11 198003 | 198003 |
| House No 90 Sec-14 154483 | 154483 |



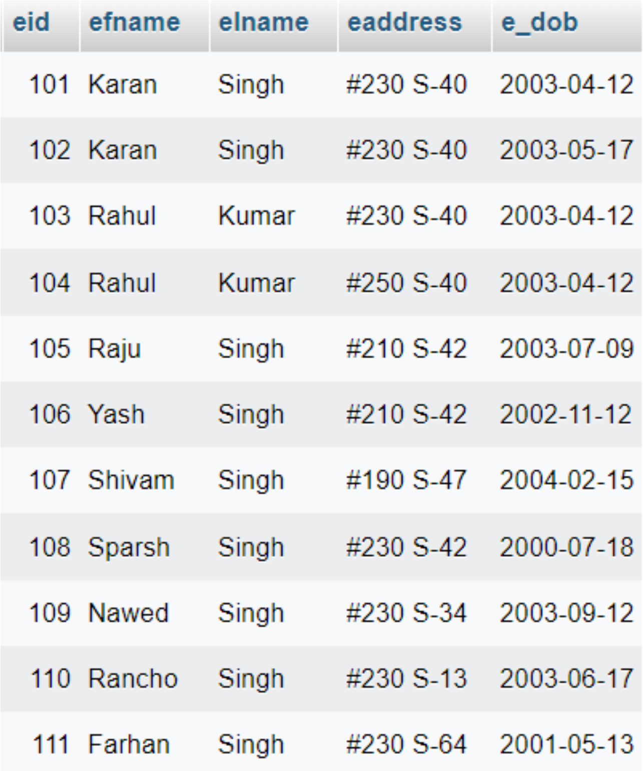
```sql
INSERT INTO customer VALUES
(1,'Shivam','Gupta','House No 12 Sec-30 121003','+91 923456xxxx'),
(2,'Shivam','Gupta','House No 10 Sec-14 121003','+91 983456xxxx'),
(3,'Yash','Kumar','House No 10 Sec-14 120034','+91 983456xxxx'),
(4,'Yash','Gupta','House No 9 Sec-11 198003','+91 943456xxxx'),
(5,'Ram','Gupta','House No 131 Sec-14 121003','+91 700456xxxx'),
(6,'Shivam','Gupta','House No 232 Sec-14 149003','+91 983456xxxx'),
(7,'Paras','Sharma','House No 546 Sec-14 132003','+91 983456xxxx'),
(8,'Pragun','Chaudhary','House No 643 Sec-14 148023','+91 983456xxxx'),
(9,'Sparsh','Singhal','House No 42 Sec-14 145483','+91 983456xxxx'),
(10,'Nawed','Singh','House No 90 Sec-14 154483','+91 983456xxxx'),
(11,'Hardik','Dhall','House No 87 Sec-14 144603','+91 983456xxxx'),
(12,'Priya','Malik','House No 878 Sec-14 151503','+91 983456xxxx')
```



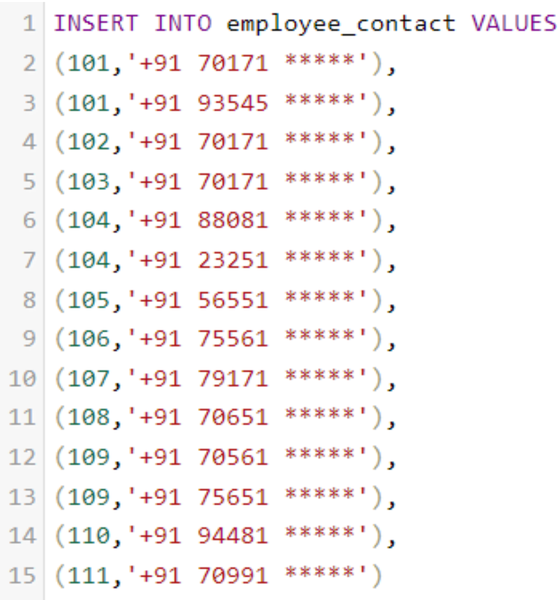
| cid | cfname | clname | caddress | c_contact |
|---|---|---|---|---|
| 1 | Shivam | Gupta | House No 12 Sec-30 121003 | +91 923456xxxx |
| 2 | Shivam | Gupta | House No 10 Sec-14 121003 | +91 983456xxxx |
| 3 | Yash | Kumar | House No 10 Sec-14 120034 | +91 983456xxxx |
| 4 | Yash | Gupta | House No 9 Sec-11 198003 | +91 943456xxxx |
| 5 | Ram | Gupta | House No 131 Sec-14 121003 | +91 700456xxxx |
| 6 | Shivam | Gupta | House No 232 Sec-14 149003 | +91 983456xxxx |
| 7 | Paras | Sharma | House No 546 Sec-14 132003 | +91 983456xxxx |
| 8 | Pragun | Chaudhary | House No 643 Sec-14 148023 | +91 983456xxxx |
| 9 | Sparsh | Singhal | House No 42 Sec-14 145483 | +91 983456xxxx |
| 10 | Nawed | Singh | House No 90 Sec-14 154483 | +91 983456xxxx |
| 11 | Hardik | Dhall | House No 87 Sec-14 144603 | +91 983456xxxx |
| 12 | Priya | Malik | House No 878 Sec-14 151503 | +91 983456xxxx |



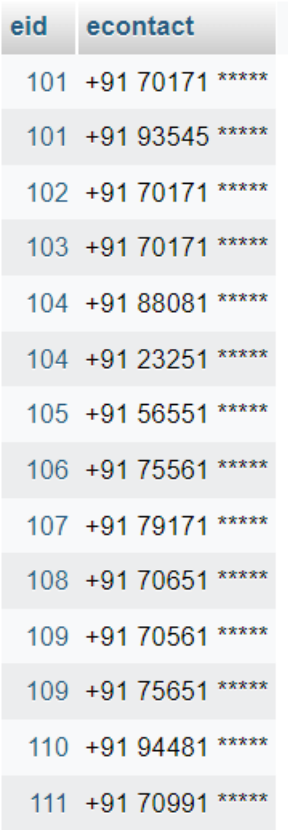
```sql
INSERT INTO employee_details VALUES
(101,'Karan','Singh','#230 S-40','2003-04-12'),
(102,'Karan','Singh','#230 S-40','2003-05-17'),
(103,'Rahul','Kumar','#230 S-40','2003-04-12'),
(104,'Rahul','Kumar','#250 S-40','2003-04-12'),
(105,'Raju','Singh','#210 S-42','2003-07-09'),
(106,'Yash','Singh','#210 S-42','2002-11-12'),
(107,'Shivam','Singh','#190 S-47','2004-02-15'),
(108,'Sparsh','Singh','#230 S-42','2000-07-18'),
(109,'Nawed','Singh','#230 S-34','2003-09-12'),
(110,'Rancho','Singh','#230 S-13','2003-06-17'),
(111,'Farhan','Singh','#230 S-64','2001-05-13')
```



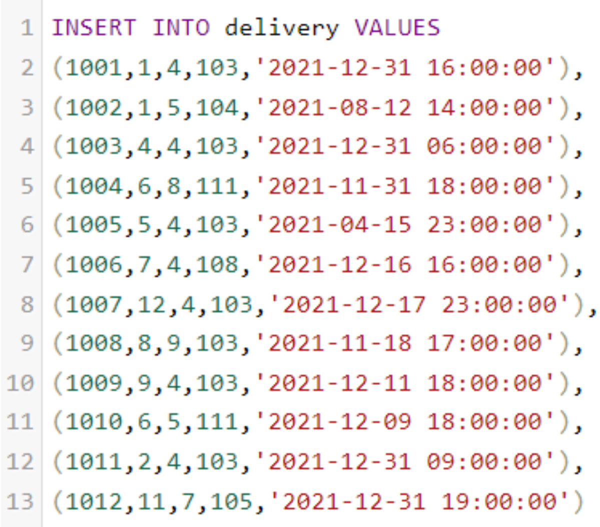
| eid | efname | elname | eaddress | e_dob |
|---|---|---|---|---|
| 101 | Karan | Singh | #230 S-40 | 2003-04-12 |
| 102 | Karan | Singh | #230 S-40 | 2003-05-17 |
| 103 | Rahul | Kumar | #230 S-40 | 2003-04-12 |
| 104 | Rahul | Kumar | #250 S-40 | 2003-04-12 |
| 105 | Raju | Singh | #210 S-42 | 2003-07-09 |
| 106 | Yash | Singh | #210 S-42 | 2002-11-12 |
| 107 | Shivam | Singh | #190 S-47 | 2004-02-15 |
| 108 | Sparsh | Singh | #230 S-42 | 2000-07-18 |
| 109 | Nawed | Singh | #230 S-34 | 2003-09-12 |
| 110 | Rancho | Singh | #230 S-13 | 2003-06-17 |
| 111 | Farhan | Singh | #230 S-64 | 2001-05-13 |



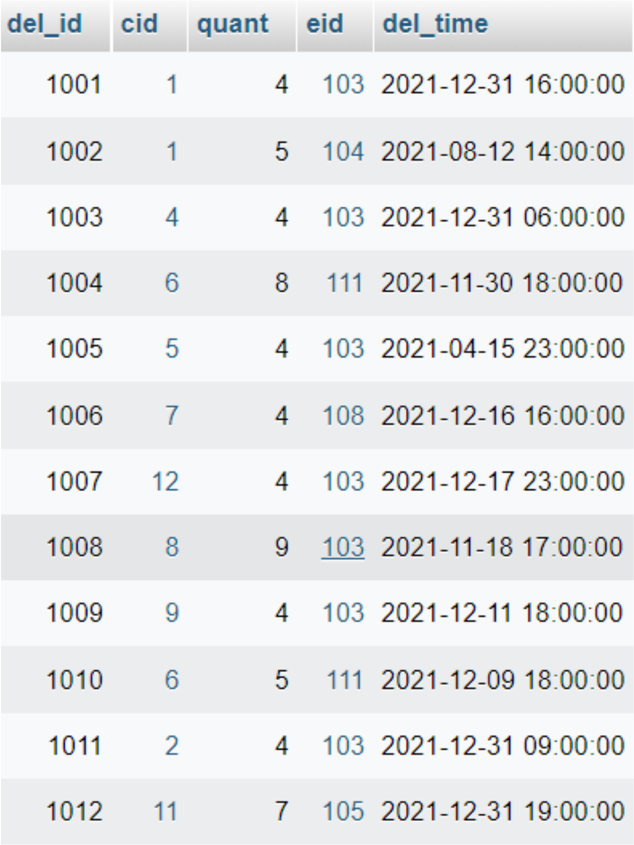
```sql
INSERT INTO employee_contact VALUES
(101,'+91 70171 *****'),
(101,'+91 93545 *****'),
(102,'+91 70171 *****'),
(103,'+91 70171 *****'),
(104,'+91 88081 *****'),
(104,'+91 23251 *****'),
(105,'+91 56551 *****'),
(106,'+91 75561 *****'),
(107,'+91 79171 *****'),
(108,'+91 70651 *****'),
(109,'+91 70561 *****'),
(109,'+91 75651 *****'),
(110,'+91 94481 *****'),
(111,'+91 70991 *****')
```



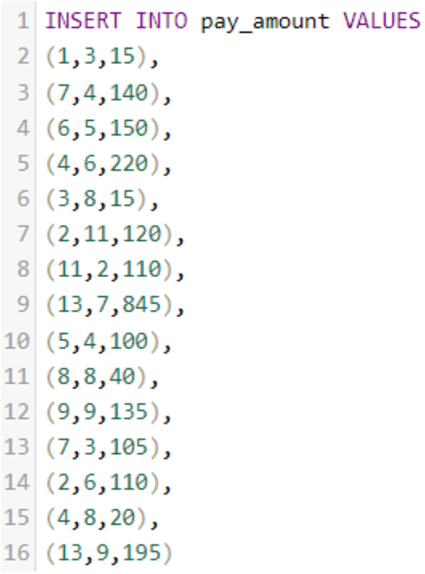
| eid | econtact |
|---|---|
| 101 | +91 70171 ***** |
| 101 | +91 93545 ***** |
| 102 | +91 70171 ***** |
| 103 | +91 70171 ***** |
| 104 | +91 88081 ***** |
| 104 | +91 23251 ***** |
| 105 | +91 56551 ***** |
| 106 | +91 75561 ***** |
| 107 | +91 79171 ***** |
| 108 | +91 70651 ***** |
| 109 | +91 70561 ***** |
| 109 | +91 75651 ***** |
| 110 | +91 94481 ***** |
| 111 | +91 70991 ***** |



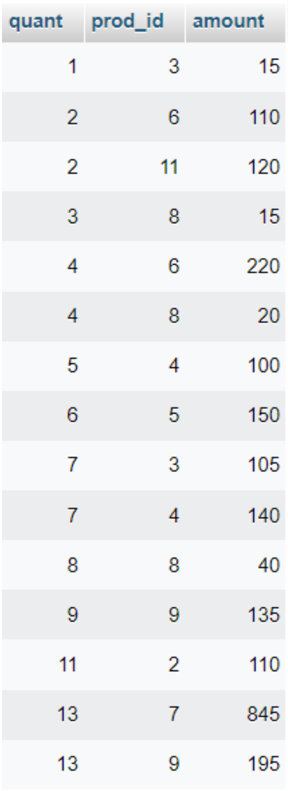
```sql
INSERT INTO delivery VALUES
(1001,1,4,103,'2021-12-31 16:00:00'),
(1002,1,5,104,'2021-08-12 14:00:00'),
(1003,4,4,103,'2021-12-31 06:00:00'),
(1004,6,8,111,'2021-11-31 18:00:00'),
(1005,5,4,103,'2021-04-15 23:00:00'),
(1006,7,4,108,'2021-12-16 16:00:00'),
(1007,12,4,103,'2021-12-17 23:00:00'),
(1008,8,9,103,'2021-11-18 17:00:00'),
(1009,9,4,103,'2021-12-11 18:00:00'),
(1010,6,5,111,'2021-12-09 18:00:00'),
(1011,2,4,103,'2021-12-31 09:00:00'),
(1012,11,7,105,'2021-12-31 19:00:00')
```



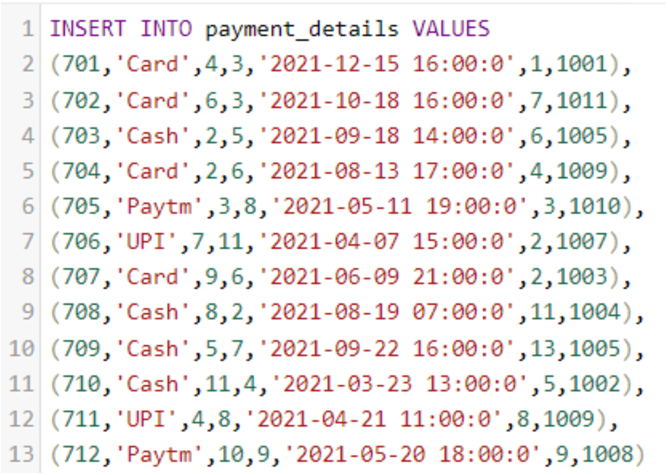
| del_id | cid | quant | eid | del_time |
|---|---|---|---|---|
| 1001 | 1 | 4 | 103 | 2021-12-31 16:00:00 |
| 1002 | 1 | 5 | 104 | 2021-08-12 14:00:00 |
| 1003 | 4 | 4 | 103 | 2021-12-31 06:00:00 |
| 1004 | 6 | 8 | 111 | 2021-11-30 18:00:00 |
| 1005 | 5 | 4 | 103 | 2021-04-15 23:00:00 |
| 1006 | 7 | 4 | 108 | 2021-12-16 16:00:00 |
| 1007 | 12 | 4 | 103 | 2021-12-17 23:00:00 |
| 1008 | 8 | 9 | 103 | 2021-11-18 17:00:00 |
| 1009 | 9 | 4 | 103 | 2021-12-11 18:00:00 |
| 1010 | 6 | 5 | 111 | 2021-12-09 18:00:00 |
| 1011 | 2 | 4 | 103 | 2021-12-31 09:00:00 |
| 1012 | 11 | 7 | 105 | 2021-12-31 19:00:00 |



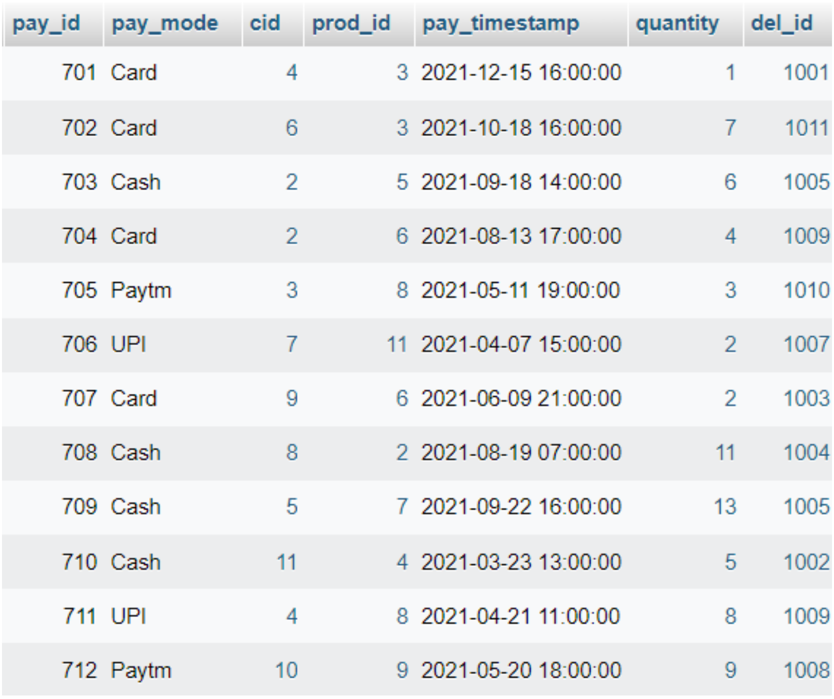
```sql
INSERT INTO pay_amount VALUES
(1,3,15),
(7,4,140),
(6,5,150),
(4,6,220),
(3,8,15),
(2,11,120),
(11,2,110),
(13,7,845),
(5,4,100),
(8,8,40),
(9,9,135),
(7,3,105),
(2,6,110),
(4,8,20),
(13,9,195)
```



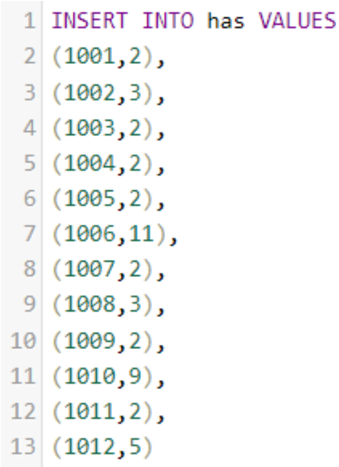
| quant | prod_id | amount |
|---|---|---|
| 1 | 3 | 15 |
| 2 | 6 | 110 |
| 2 | 11 | 120 |
| 3 | 8 | 15 |
| 4 | 6 | 220 |
| 4 | 8 | 20 |
| 5 | 4 | 100 |
| 6 | 5 | 150 |
| 7 | 3 | 105 |
| 7 | 4 | 140 |
| 8 | 8 | 40 |
| 9 | 9 | 135 |
| 11 | 2 | 110 |
| 13 | 7 | 845 |
| 13 | 9 | 195 |



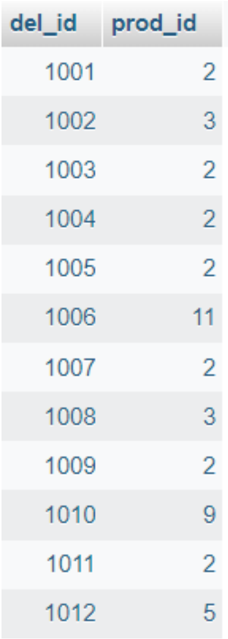
```sql
INSERT INTO payment_details VALUES
(701,'Card',4,3,'2021-12-15 16:00:0',1,1001),
(702,'Card',6,3,'2021-10-18 16:00:0',7,1011),
(703,'Cash',2,5,'2021-09-18 14:00:0',6,1005),
(704,'Card',2,6,'2021-08-13 17:00:0',4,1009),
(705,'Paytm',3,8,'2021-05-11 19:00:0',3,1010),
(706,'UPI',7,11,'2021-04-07 15:00:0',2,1007),
(707,'Card',9,6,'2021-06-09 21:00:0',2,1003),
(708,'Cash',8,2,'2021-08-19 07:00:0',11,1004),
(709,'Cash',5,7,'2021-09-22 16:00:0',13,1005),
(710,'Cash',11,4,'2021-03-23 13:00:0',5,1002),
(711,'UPI',4,8,'2021-04-21 11:00:0',8,1009),
(712,'Paytm',10,9,'2021-05-20 18:00:0',9,1008)
```



| pay_id | pay_mode | cid | prod_id | pay_timestamp | quantity | del_id |
|---|---|---|---|---|---|---|
| 701 | Card | 4 | 3 | 2021-12-15 16:00:00 | 1 | 1001 |
| 702 | Card | 6 | 3 | 2021-10-18 16:00:00 | 7 | 1011 |
| 703 | Cash | 2 | 5 | 2021-09-18 14:00:00 | 6 | 1005 |
| 704 | Card | 2 | 6 | 2021-08-13 17:00:00 | 4 | 1009 |
| 705 | Paytm | 3 | 8 | 2021-05-11 19:00:00 | 3 | 1010 |
| 706 | UPI | 7 | 11 | 2021-04-07 15:00:00 | 2 | 1007 |
| 707 | Card | 9 | 6 | 2021-06-09 21:00:00 | 2 | 1003 |
| 708 | Cash | 8 | 2 | 2021-08-19 07:00:00 | 11 | 1004 |
| 709 | Cash | 5 | 7 | 2021-09-22 16:00:00 | 13 | 1005 |
| 710 | Cash | 11 | 4 | 2021-03-23 13:00:00 | 5 | 1002 |
| 711 | UPI | 4 | 8 | 2021-04-21 11:00:00 | 8 | 1009 |
| 712 | Paytm | 10 | 9 | 2021-05-20 18:00:00 | 9 | 1008 |



```sql
INSERT INTO has VALUES
(1001,2),
(1002,3),
(1003,2),
(1004,2),
(1005,2),
(1006,11),
(1007,2),
(1008,3),
(1009,2),
(1010,9),
(1011,2),
(1012,5)
```

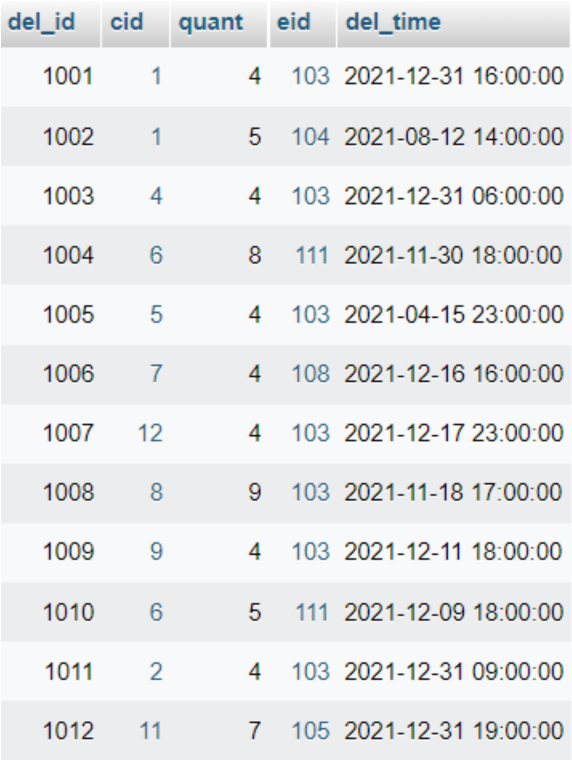


| del_id | prod_id |
|---|---|
| 1001 | 2 |
| 1002 | 3 |
| 1003 | 2 |
| 1004 | 2 |
| 1005 | 2 |
| 1006 | 11 |
| 1007 | 2 |
| 1008 | 3 |
| 1009 | 2 |
| 1010 | 9 |
| 1011 | 2 |
| 1012 | 5 |

# *Updation Queries*

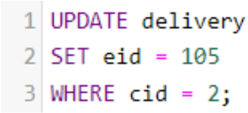
1) Write a query to update a specific delivery partner associated with a particular customer id

## Before Update



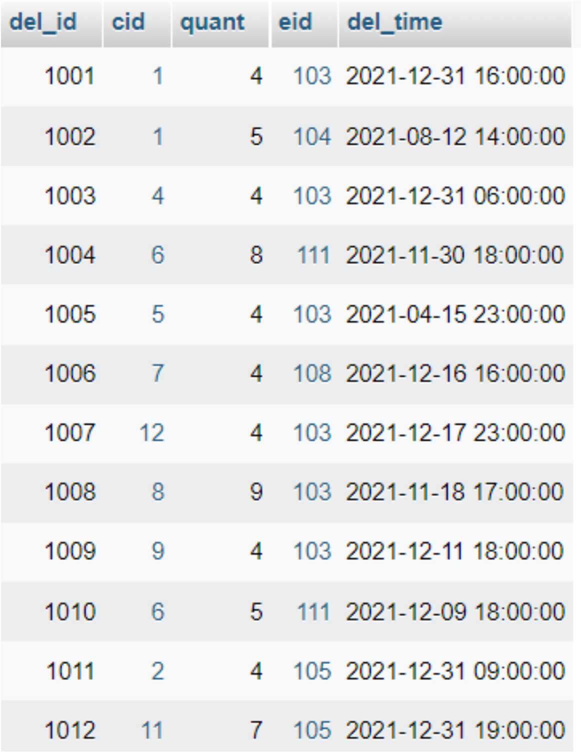
| del_id | cid | quant | eid | del_time |
|---|---|---|---|---|
| 1001 | 1 | 4 | 103 | 2021-12-31 16:00:00 |
| 1002 | 1 | 5 | 104 | 2021-08-12 14:00:00 |
| 1003 | 4 | 4 | 103 | 2021-12-31 06:00:00 |
| 1004 | 6 | 8 | 111 | 2021-11-30 18:00:00 |
| 1005 | 5 | 4 | 103 | 2021-04-15 23:00:00 |
| 1006 | 7 | 4 | 108 | 2021-12-16 16:00:00 |
| 1007 | 12 | 4 | 103 | 2021-12-17 23:00:00 |
| 1008 | 8 | 9 | 103 | 2021-11-18 17:00:00 |
| 1009 | 9 | 4 | 103 | 2021-12-11 18:00:00 |
| 1010 | 6 | 5 | 111 | 2021-12-09 18:00:00 |
| 1011 | 2 | 4 | 103 | 2021-12-31 09:00:00 |
| 1012 | 11 | 7 | 105 | 2021-12-31 19:00:00 |

## Query



```sql
UPDATE delivery
SET eid = 105
WHERE cid = 2;
```

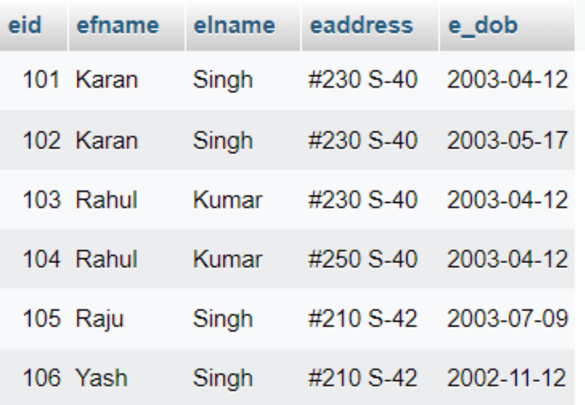
## After Update

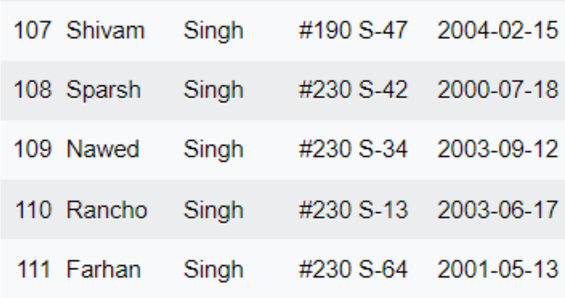


| del_id | cid | quant | eid | del_time |
|---|---|---|---|---|
| 1001 | 1 | 4 | 103 | 2021-12-31 16:00:00 |
| 1002 | 1 | 5 | 104 | 2021-08-12 14:00:00 |
| 1003 | 4 | 4 | 103 | 2021-12-31 06:00:00 |
| 1004 | 6 | 8 | 111 | 2021-11-30 18:00:00 |
| 1005 | 5 | 4 | 103 | 2021-04-15 23:00:00 |
| 1006 | 7 | 4 | 108 | 2021-12-16 16:00:00 |
| 1007 | 12 | 4 | 103 | 2021-12-17 23:00:00 |
| 1008 | 8 | 9 | 103 | 2021-11-18 17:00:00 |
| 1009 | 9 | 4 | 103 | 2021-12-11 18:00:00 |
| 1010 | 6 | 5 | 111 | 2021-12-09 18:00:00 |
| 1011 | 2 | 4 | 105 | 2021-12-31 09:00:00 |
| 1012 | 11 | 7 | 105 | 2021-12-31 19:00:00 |

2) Write a query to update a address of a specific

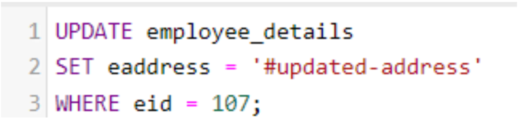
employeeBefore Update





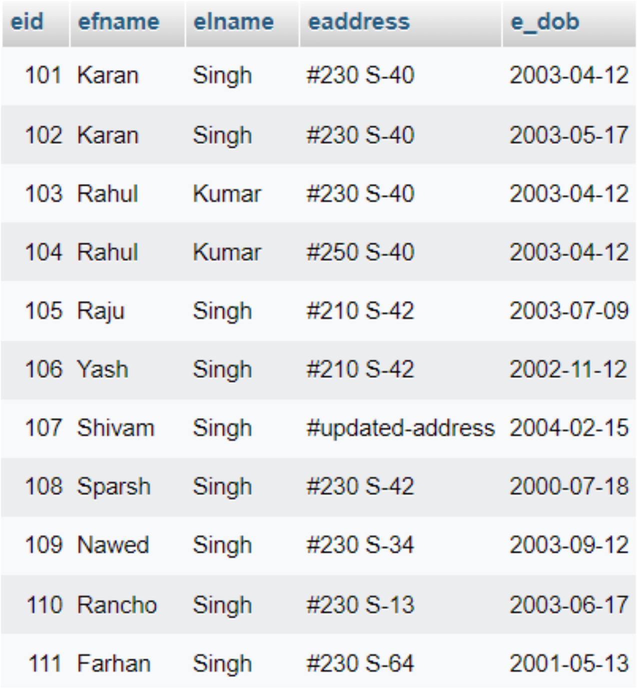
| eid | efname | elname | eaddress | e_dob |
|---|---|---|---|---|
| 101 | Karan | Singh | #230 S-40 | 2003-04-12 |
| 102 | Karan | Singh | #230 S-40 | 2003-05-17 |
| 103 | Rahul | Kumar | #230 S-40 | 2003-04-12 |
| 104 | Rahul | Kumar | #250 S-40 | 2003-04-12 |
| 105 | Raju | Singh | #210 S-42 | 2003-07-09 |
| 106 | Yash | Singh | #210 S-42 | 2002-11-12 |
| 107 | Shivam | Singh | #190 S-47 | 2004-02-15 |
| 108 | Sparsh | Singh | #230 S-42 | 2000-07-18 |
| 109 | Nawed | Singh | #230 S-34 | 2003-09-12 |
| 110 | Rancho | Singh | #230 S-13 | 2003-06-17 |
| 111 | Farhan | Singh | #230 S-64 | 2001-05-13 |

## Query



```
UPDATE employee_details
SET eaddress = '#updated-address'
WHERE eid = 107;
```

## After Update

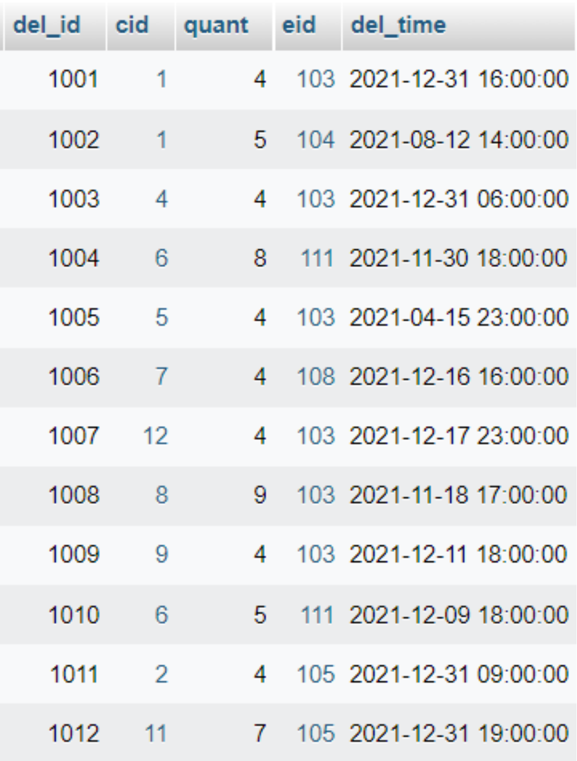


| eid | efname | elname | eaddress | e_dob |
|---|---|---|---|---|
| 101 | Karan | Singh | #230 S-40 | 2003-04-12 |
| 102 | Karan | Singh | #230 S-40 | 2003-05-17 |
| 103 | Rahul | Kumar | #230 S-40 | 2003-04-12 |
| 104 | Rahul | Kumar | #250 S-40 | 2003-04-12 |
| 105 | Raju | Singh | #210 S-42 | 2003-07-09 |
| 106 | Yash | Singh | #210 S-42 | 2002-11-12 |
| 107 | Shivam | Singh | #updated-address | 2004-02-15 |
| 108 | Sparsh | Singh | #230 S-42 | 2000-07-18 |
| 109 | Nawed | Singh | #230 S-34 | 2003-09-12 |
| 110 | Rancho | Singh | #230 S-13 | 2003-06-17 |
| 111 | Farhan | Singh | #230 S-64 | 2001-05-13 |

## *Deletion Queries*

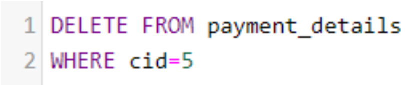
1) Assume that a payment is cancelled for a particular customer write a query to delete that entry

### Before Delete



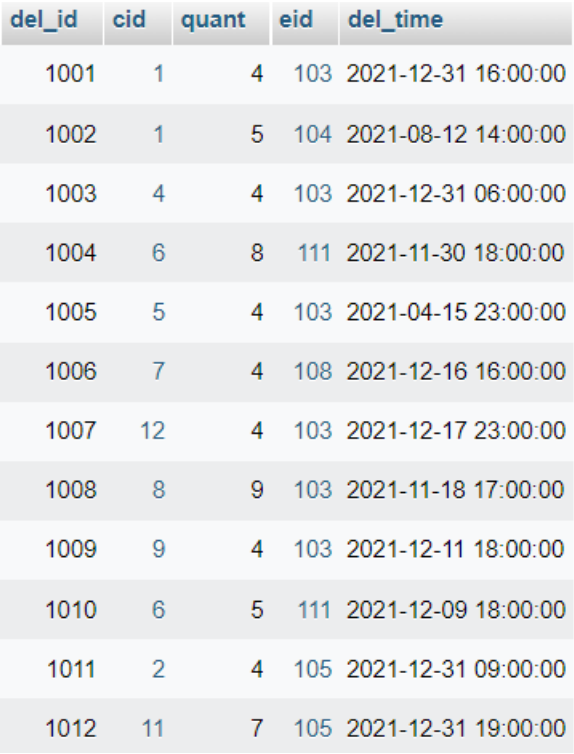
| del_id | cid | quant | eid | del_time |
|---|---|---|---|---|
| 1001 | 1 | 4 | 103 | 2021-12-31 16:00:00 |
| 1002 | 1 | 5 | 104 | 2021-08-12 14:00:00 |
| 1003 | 4 | 4 | 103 | 2021-12-31 06:00:00 |
| 1004 | 6 | 8 | 111 | 2021-11-30 18:00:00 |
| 1005 | 5 | 4 | 103 | 2021-04-15 23:00:00 |
| 1006 | 7 | 4 | 108 | 2021-12-16 16:00:00 |
| 1007 | 12 | 4 | 103 | 2021-12-17 23:00:00 |
| 1008 | 8 | 9 | 103 | 2021-11-18 17:00:00 |
| 1009 | 9 | 4 | 103 | 2021-12-11 18:00:00 |
| 1010 | 6 | 5 | 111 | 2021-12-09 18:00:00 |
| 1011 | 2 | 4 | 105 | 2021-12-31 09:00:00 |
| 1012 | 11 | 7 | 105 | 2021-12-31 19:00:00 |

### Query



```
DELETE FROM payment_details
WHERE cid=5
```

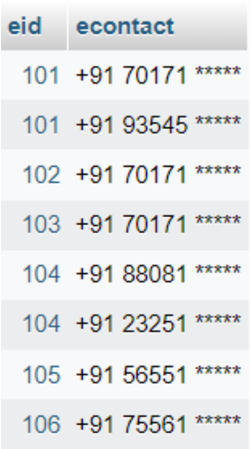
### After Delete

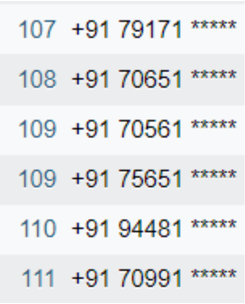


| del_id | cid | quant | eid | del_time |
|---|---|---|---|---|
| 1001 | 1 | 4 | 103 | 2021-12-31 16:00:00 |
| 1002 | 1 | 5 | 104 | 2021-08-12 14:00:00 |
| 1003 | 4 | 4 | 103 | 2021-12-31 06:00:00 |
| 1004 | 6 | 8 | 111 | 2021-11-30 18:00:00 |
| 1005 | 5 | 4 | 103 | 2021-04-15 23:00:00 |
| 1006 | 7 | 4 | 108 | 2021-12-16 16:00:00 |
| 1007 | 12 | 4 | 103 | 2021-12-17 23:00:00 |
| 1008 | 8 | 9 | 103 | 2021-11-18 17:00:00 |
| 1009 | 9 | 4 | 103 | 2021-12-11 18:00:00 |
| 1010 | 6 | 5 | 111 | 2021-12-09 18:00:00 |
| 1011 | 2 | 4 | 105 | 2021-12-31 09:00:00 |
| 1012 | 11 | 7 | 105 | 2021-12-31 19:00:00 |

2) Assume that a employee want to leave for organisation write a query to delete the contact of that employee

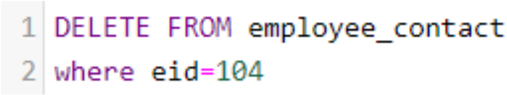
### Before Delete





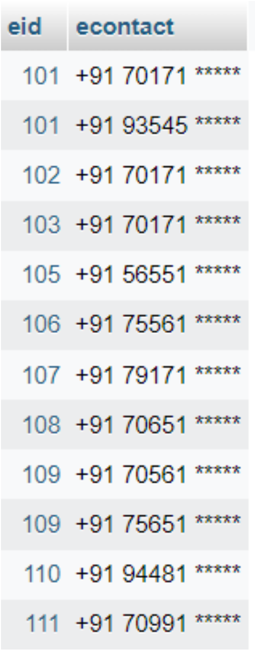
| eid | econtact |
|---|---|
| 101 | +91 70171 ***** |
| 101 | +91 93545 ***** |
| 102 | +91 70171 ***** |
| 103 | +91 70171 ***** |
| 104 | +91 88081 ***** |
| 104 | +91 23251 ***** |
| 105 | +91 56551 ***** |
| 106 | +91 75561 ***** |
| 107 | +91 79171 ***** |
| 108 | +91 70651 ***** |
| 109 | +91 70561 ***** |
| 109 | +91 75651 ***** |
| 110 | +91 94481 ***** |
| 111 | +91 70991 ***** |

## Query



```
DELETE FROM employee_contact
where eid=104
```

## After Update

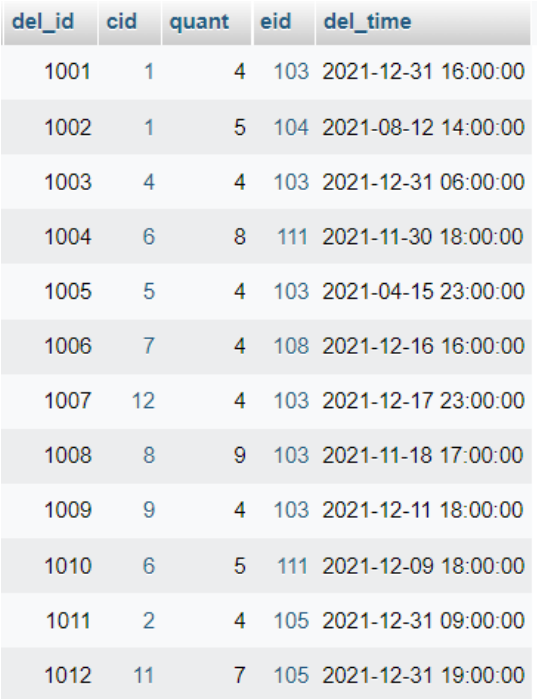


| eid | econtact |
|---|---|
| 101 | +91 70171 ***** |
| 101 | +91 93545 ***** |
| 102 | +91 70171 ***** |
| 103 | +91 70171 ***** |
| 105 | +91 56551 ***** |
| 106 | +91 75561 ***** |
| 107 | +91 79171 ***** |
| 108 | +91 70651 ***** |
| 109 | +91 70561 ***** |
| 109 | +91 75651 ***** |
| 110 | +91 94481 ***** |
| 111 | +91 70991 ***** |

# *Schema Alter Queries*

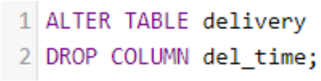
1) Database designer of your company wants that delivery timestamp should be deleted write a query to do so

Before Alter



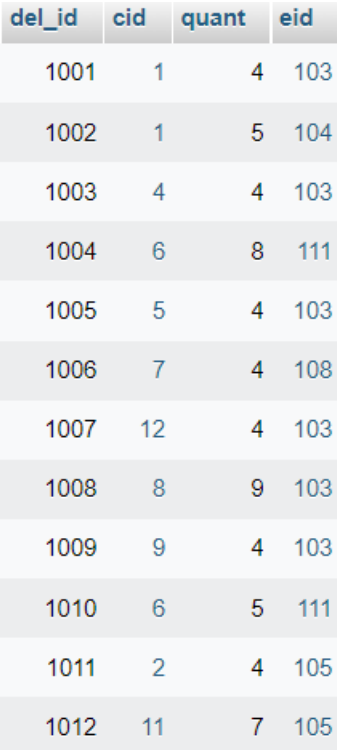
| del_id | cid | quant | eid | del_time |
|---|---|---|---|---|
| 1001 | 1 | 4 | 103 | 2021-12-31 16:00:00 |
| 1002 | 1 | 5 | 104 | 2021-08-12 14:00:00 |
| 1003 | 4 | 4 | 103 | 2021-12-31 06:00:00 |
| 1004 | 6 | 8 | 111 | 2021-11-30 18:00:00 |
| 1005 | 5 | 4 | 103 | 2021-04-15 23:00:00 |
| 1006 | 7 | 4 | 108 | 2021-12-16 16:00:00 |
| 1007 | 12 | 4 | 103 | 2021-12-17 23:00:00 |
| 1008 | 8 | 9 | 103 | 2021-11-18 17:00:00 |
| 1009 | 9 | 4 | 103 | 2021-12-11 18:00:00 |
| 1010 | 6 | 5 | 111 | 2021-12-09 18:00:00 |
| 1011 | 2 | 4 | 105 | 2021-12-31 09:00:00 |
| 1012 | 11 | 7 | 105 | 2021-12-31 19:00:00 |

Query



```
ALTER TABLE delivery
DROP COLUMN del_time;
```

After Alter

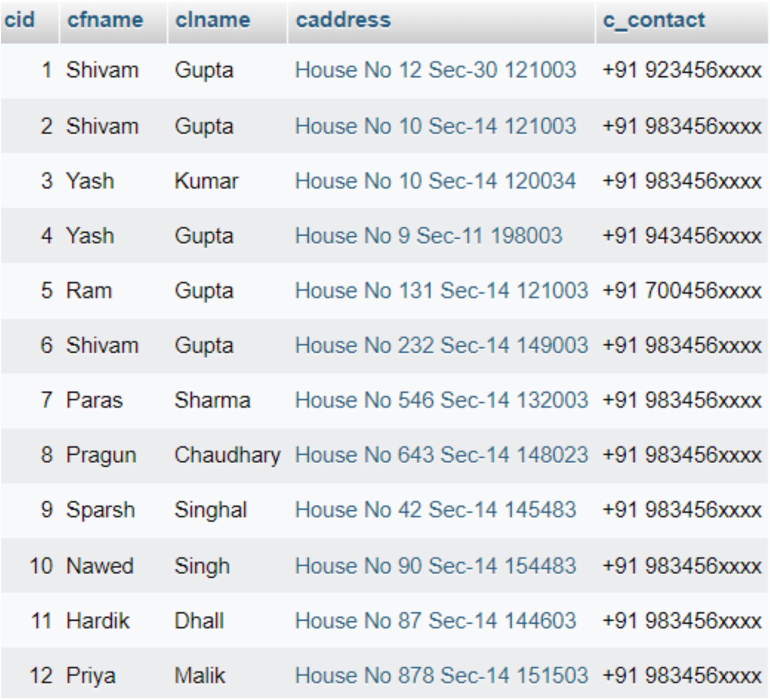


| del_id | cid | quant | eid |
|---|---|---|---|
| 1001 | 1 | 4 | 103 |
| 1002 | 1 | 5 | 104 |
| 1003 | 4 | 4 | 103 |
| 1004 | 6 | 8 | 111 |
| 1005 | 5 | 4 | 103 |
| 1006 | 7 | 4 | 108 |
| 1007 | 12 | 4 | 103 |
| 1008 | 8 | 9 | 103 |
| 1009 | 9 | 4 | 103 |
| 1010 | 6 | 5 | 111 |
| 1011 | 2 | 4 | 105 |
| 1012 | 11 | 7 | 105 |

2) Database designer of your company wants that AGE OF YOUR customer should also beadded to database write a query to do so

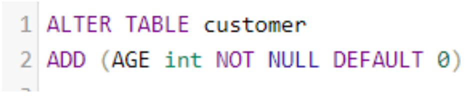
Before Alter

**PTO**



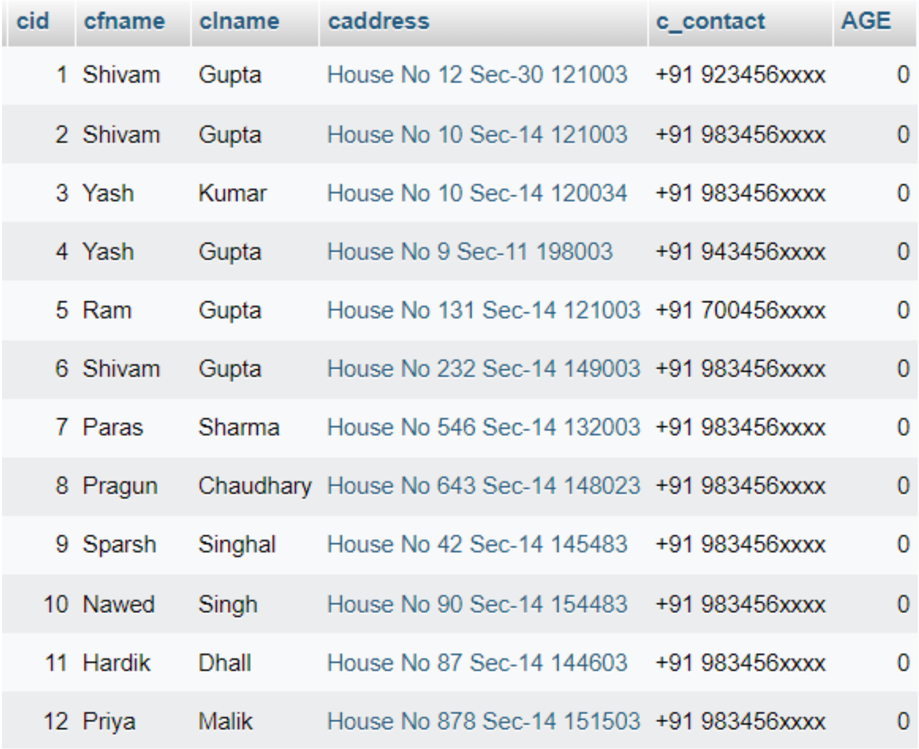
| cid | cfname | clname | caddress | c_contact |
|---|---|---|---|---|
| 1 | Shivam | Gupta | House No 12 Sec-30 121003 | +91 923456xxxx |
| 2 | Shivam | Gupta | House No 10 Sec-14 121003 | +91 983456xxxx |
| 3 | Yash | Kumar | House No 10 Sec-14 120034 | +91 983456xxxx |
| 4 | Yash | Gupta | House No 9 Sec-11 198003 | +91 943456xxxx |
| 5 | Ram | Gupta | House No 131 Sec-14 121003 | +91 700456xxxx |
| 6 | Shivam | Gupta | House No 232 Sec-14 149003 | +91 983456xxxx |
| 7 | Paras | Sharma | House No 546 Sec-14 132003 | +91 983456xxxx |
| 8 | Pragun | Chaudhary | House No 643 Sec-14 148023 | +91 983456xxxx |
| 9 | Sparsh | Singhal | House No 42 Sec-14 145483 | +91 983456xxxx |
| 10 | Nawed | Singh | House No 90 Sec-14 154483 | +91 983456xxxx |
| 11 | Hardik | Dhall | House No 87 Sec-14 144603 | +91 983456xxxx |
| 12 | Priya | Malik | House No 878 Sec-14 151503 | +91 983456xxxx |

## Query



```
ALTER TABLE customer
ADD (AGE int NOT NULL DEFAULT 0)
```

## After Alter

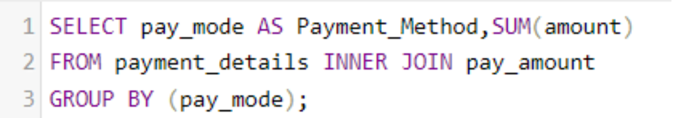


| cid | cfname | clname | caddress | c_contact | AGE |
|---|---|---|---|---|---|
| 1 | Shivam | Gupta | House No 12 Sec-30 121003 | +91 923456xxxx | 0 |
| 2 | Shivam | Gupta | House No 10 Sec-14 121003 | +91 983456xxxx | 0 |
| 3 | Yash | Kumar | House No 10 Sec-14 120034 | +91 983456xxxx | 0 |
| 4 | Yash | Gupta | House No 9 Sec-11 198003 | +91 943456xxxx | 0 |
| 5 | Ram | Gupta | House No 131 Sec-14 121003 | +91 700456xxxx | 0 |
| 6 | Shivam | Gupta | House No 232 Sec-14 149003 | +91 983456xxxx | 0 |
| 7 | Paras | Sharma | House No 546 Sec-14 132003 | +91 983456xxxx | 0 |
| 8 | Pragun | Chaudhary | House No 643 Sec-14 148023 | +91 983456xxxx | 0 |
| 9 | Sparsh | Singhal | House No 42 Sec-14 145483 | +91 983456xxxx | 0 |
| 10 | Nawed | Singh | House No 90 Sec-14 154483 | +91 983456xxxx | 0 |
| 11 | Hardik | Dhall | House No 87 Sec-14 144603 | +91 983456xxxx | 0 |
| 12 | Priya | Malik | House No 878 Sec-14 151503 | +91 983456xxxx | 0 |

# *Aggregate Function Queries*

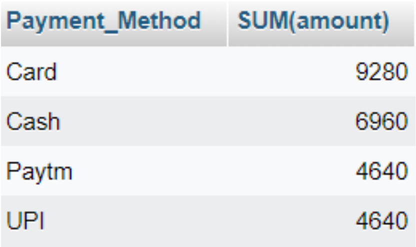
1) Write a query to find the total sum of payments for a particular mode

## Query



```
SELECT pay_mode AS Payment_Method,SUM(amount)
FROM payment_details INNER JOIN pay_amount
GROUP BY (pay_mode);
```

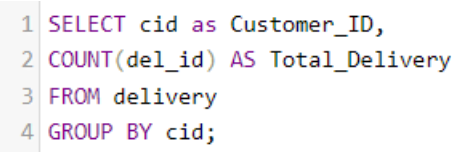
## Output



| Payment_Method | SUM(amount) |
|---|---|
| Card | 9280 |
| Cash | 6960 |
| Paytm | 4640 |
| UPI | 4640 |

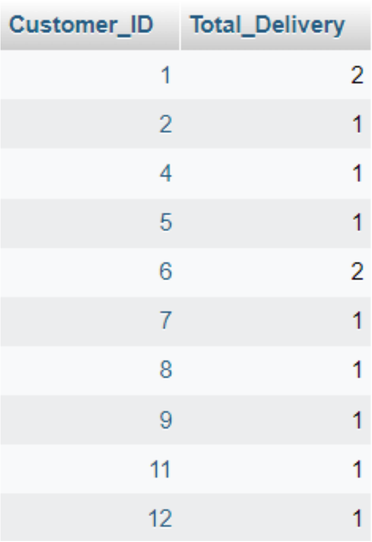
2) Write a query to count all the deliveries for a particular customer

## Query

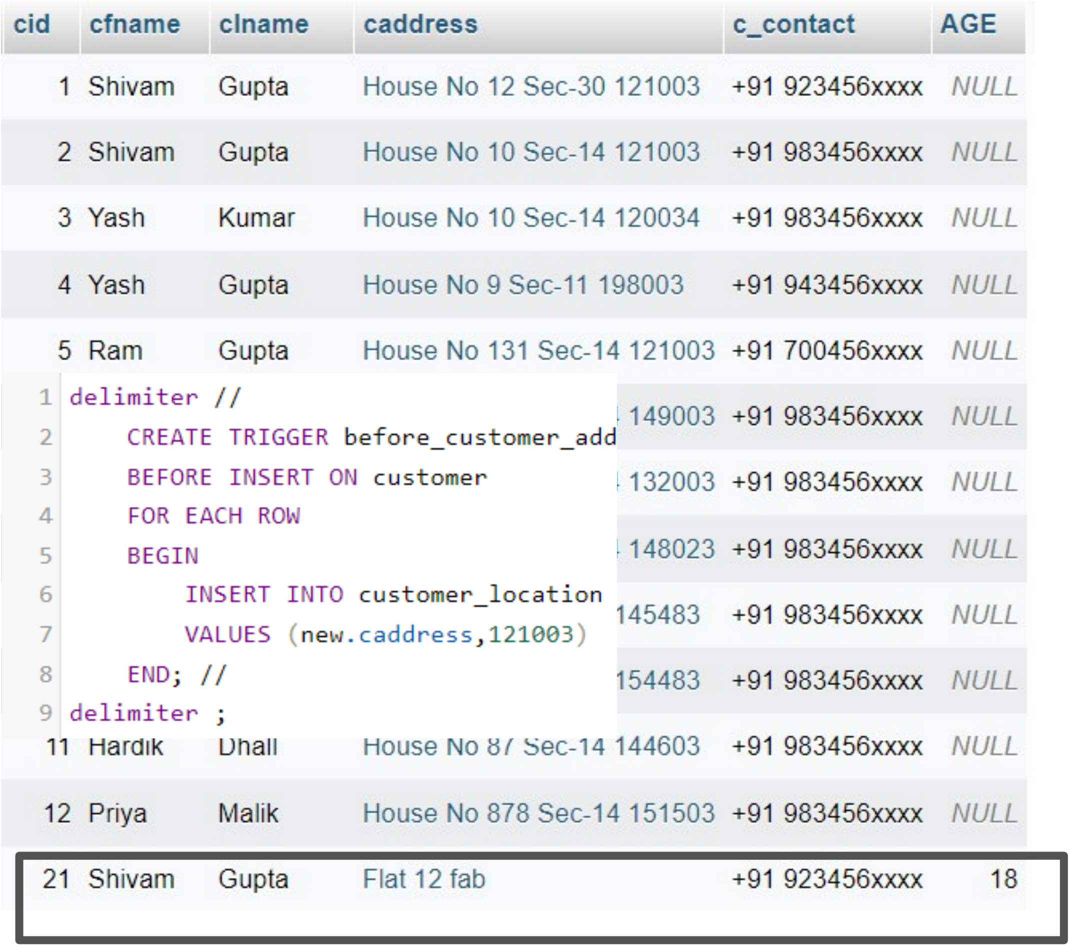


```
SELECT cid as Customer_ID,
COUNT(del_id) AS Total_Delivery
FROM delivery
GROUP BY cid;
```

## Output



| Customer_ID | Total_Delivery |
|---|---|
| 1 | 2 |
| 2 | 1 |
| 4 | 1 |
| 5 | 1 |
| 6 | 2 |
| 7 | 1 |
| 8 | 1 |
| 9 | 1 |
| 11 | 1 |
| 12 | 1 |



| cid | cfname | clname | caddress | c_contact | AGE |
|---|---|---|---|---|---|
| 1 | Shivam | Gupta | House No 12 Sec-30 121003 | +91 923456xxxx | NULL |
| 2 | Shivam | Gupta | House No 10 Sec-14 121003 | +91 983456xxxx | NULL |
| 3 | Yash | Kumar | House No 10 Sec-14 120034 | +91 983456xxxx | NULL |
| 4 | Yash | Gupta | House No 9 Sec-11 198003 | +91 943456xxxx | NULL |
| 5 | Ram | Gupta | House No 131 Sec-14 121003 | +91 700456xxxx | NULL |
|  |  |  | 149003 | +91 983456xxxx | NULL |
|  |  |  | 132003 | +91 983456xxxx | NULL |
|  |  |  | 148023 | +91 983456xxxx | NULL |
|  |  |  | 145483 | +91 983456xxxx | NULL |
|  |  |  | 154483 | +91 983456xxxx | NULL |
| 11 | Hardik | Dhall | House No 87 Sec-14 144603 | +91 983456xxxx | NULL |
| 12 | Priya | Malik | House No 878 Sec-14 151503 | +91 983456xxxx | NULL |
| 21 | Shivam | Gupta | Flat 12 fab | +91 923456xxxx | 18 |

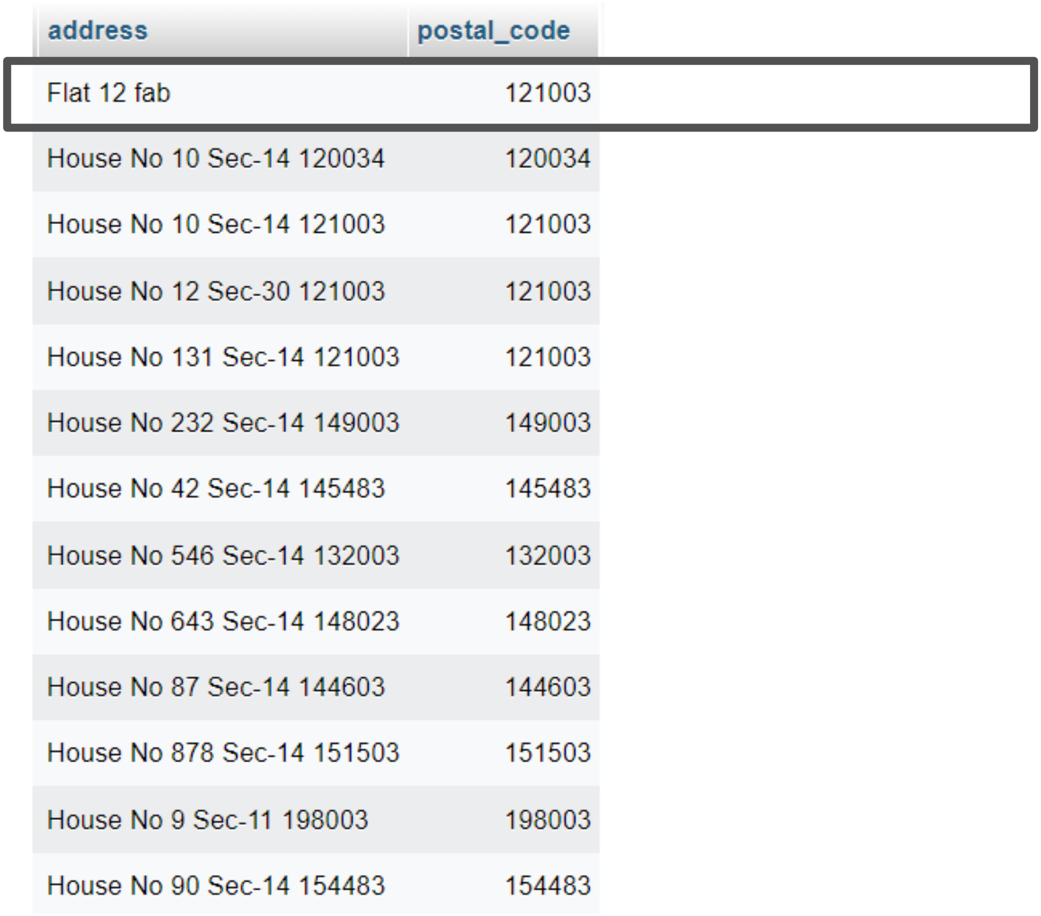
```
delimiter //
    CREATE TRIGGER before_customer_add
    BEFORE INSERT ON customer
    FOR EACH ROW
    BEGIN
        INSERT INTO customer_location
        VALUES (new.caddress,121003)
    END; //
delimiter ;
```

## *Trigger Create Queries*

1) Whenever a customer record is added a address is added tocustomer_address table with default postal code

Query

Output

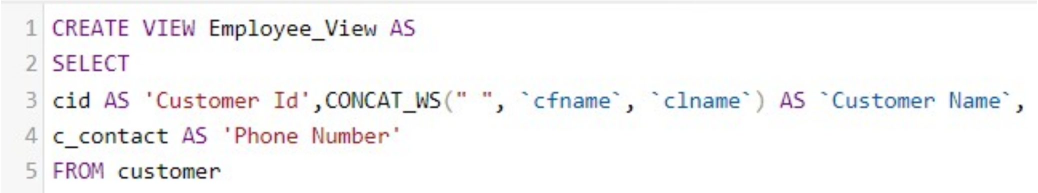


| address | postal_code |
|---|---|
| Flat 12 fab | 121003 |
| House No 10 Sec-14 120034 | 120034 |
| House No 10 Sec-14 121003 | 121003 |
| House No 12 Sec-30 121003 | 121003 |
| House No 131 Sec-14 121003 | 121003 |
| House No 232 Sec-14 149003 | 149003 |
| House No 42 Sec-14 145483 | 145483 |
| House No 546 Sec-14 132003 | 132003 |
| House No 643 Sec-14 148023 | 148023 |
| House No 87 Sec-14 144603 | 144603 |
| House No 878 Sec-14 151503 | 151503 |
| House No 9 Sec-11 198003 | 198003 |
| House No 90 Sec-14 154483 | 154483 |

# *View Create Query*

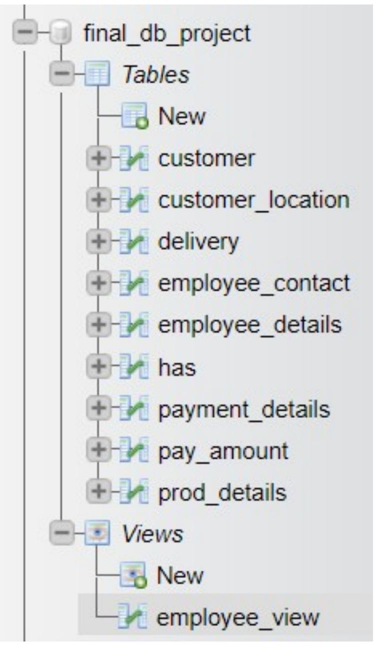
1) Create a view for an employee that holds details of customer i.e his/her name, id, phone number

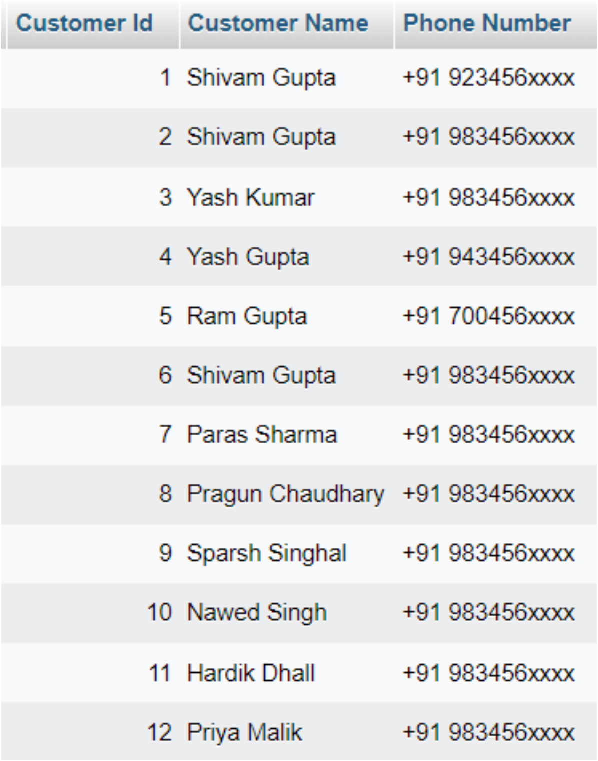
## Query



```
CREATE VIEW Employee_View AS
SELECT
cid AS 'Customer Id',CONCAT_WS(" ", `cfname`, `clname`) AS `Customer Name`,
c_contact AS 'Phone Number'
FROM customer
```

## Output





| Customer Id | Customer Name | Phone Number |
|---|---|---|
| 1 | Shivam Gupta | +91 923456xxxx |
| 2 | Shivam Gupta | +91 983456xxxx |
| 3 | Yash Kumar | +91 983456xxxx |
| 4 | Yash Gupta | +91 943456xxxx |
| 5 | Ram Gupta | +91 700456xxxx |
| 6 | Shivam Gupta | +91 983456xxxx |
| 7 | Paras Sharma | +91 983456xxxx |
| 8 | Pragun Chaudhary | +91 983456xxxx |
| 9 | Sparsh Singhal | +91 983456xxxx |
| 10 | Nawed Singh | +91 983456xxxx |
| 11 | Hardik Dhall | +91 983456xxxx |
| 12 | Priya Malik | +91 983456xxxx |