PROJECT REPORT

Group No. : 14

Member Details :

Shailesh Jadhav (Leader)	202112051
Urja Gandhi	202112039
Sharvi Gabani	202112066
Jay Bhatt	202112108
Sagar Variya	202112114

Project Description

The objective of this project is to develop an online grocery store where any grocery products can be bought from the comfort of home through the Internet. This system will be a mediator between distributors and customers.

The system is divided into two parts :

- The data of distributors and branches handled by them and stock availability is collected.
- The products are displayed to customers according to the availability, and data related to their orders is stored.

There are three roles available :

- Admin
- Distributor
- Customer

Distributor

- For every branch, there is a distributor. Stores have many grocery items, with some available stocks maintained by the branch distributor.
- Distributors will be able to view and respond to their orders. They can also view customer details and addresses, sales, and available stock of items for a particular branch.

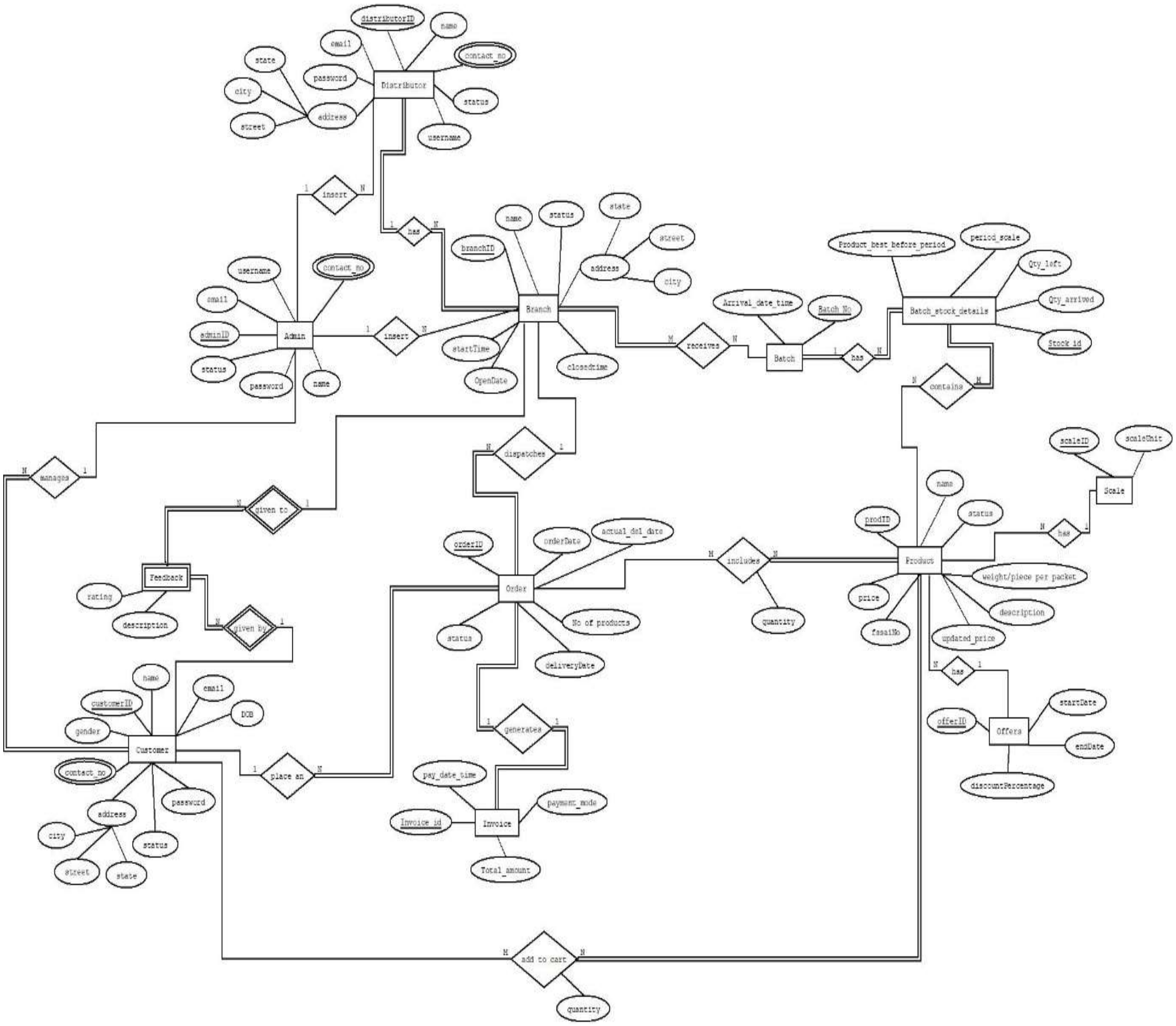
Admin

- can add products, edit products, update and remove products.
- can block a distributor or a customer if required.
- can view the branch details and sales, customer reviews, stock reports.

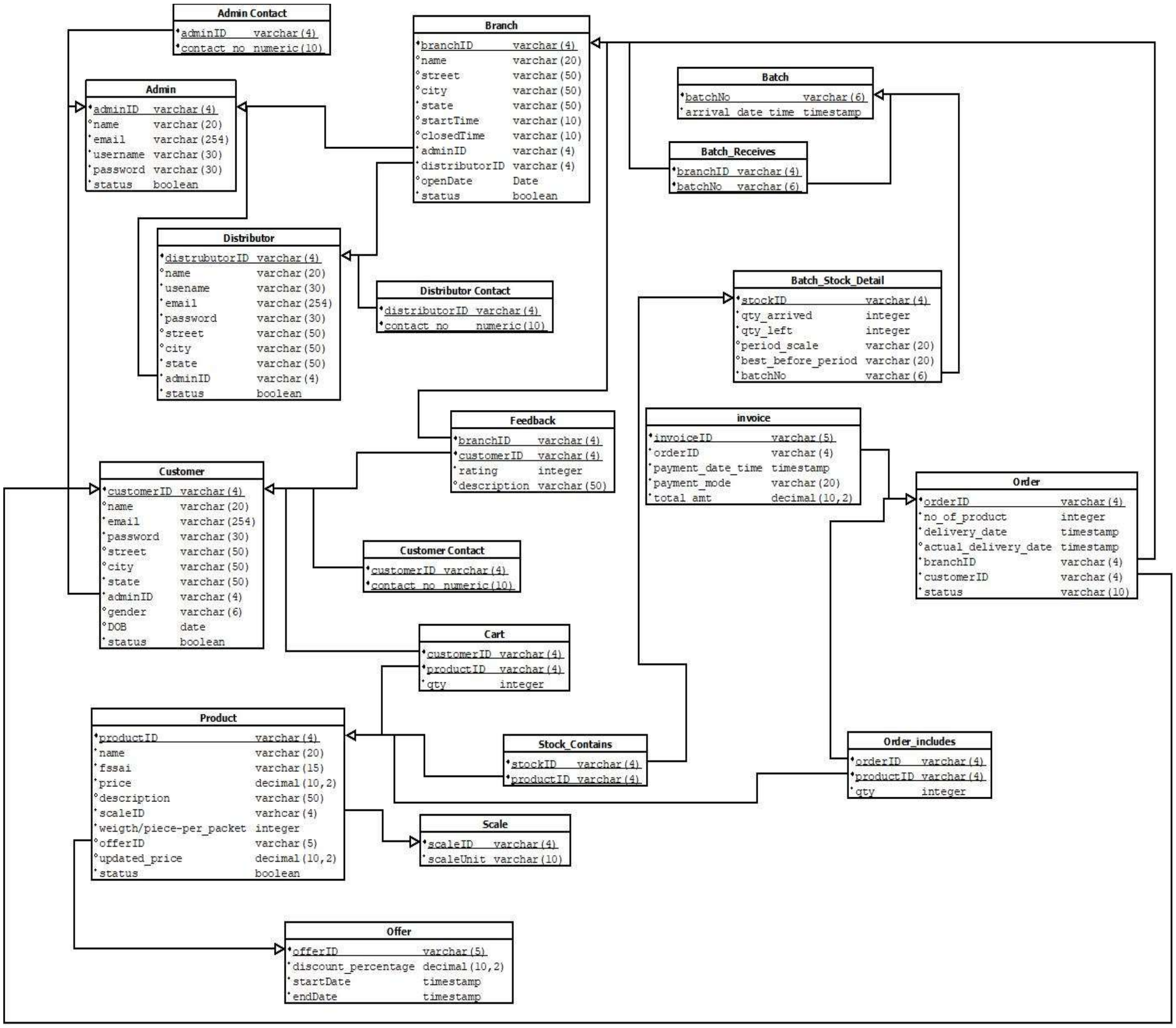
Customer

- In this virtual store, customers can browse the catalog and select products of interest. The selected items may be collected in a shopping cart.
- A customer will register or sign-up for features like purchasing products, adding to cart, order history, give feedback.
- Customers can view and order the grocery items from any available branch in their area and be able to check his/her past orders and can give feedback for that branch. He/she can even cancel the placed order.

E-R Diagram :


distributorID
name
email
contact_no
state
password
Distributor
city
address
status
street
username
insert
1
N
has
1
N
name
status
state
branchID
street
address
city
username
contact_no
Branch
email
adminID
Admin
1
insert
N
startTime
closedtime
status
password
name
OpenDate
Product_best_before_period
period_scale
Qty_left
Arrival_date_time
Batch_No
Batch_stock_details
Qty_arrived
M
receives
N
Batch
1
has
N
Stock_id
N
contains
M
N
manages
1
N
dispatches
1
scaleID
scaleUnit
N
given to
1
Scale
name
orderID
orderDate
actual_del_date
prodID
status
N
has
1
M
includes
N
Product
Feedback
Order
rating
weight/piece per packet
description
price
N
given by
1
status
No of products
quantity
description
fssaiNo
updated_price
deliveryDate
name
email
N
has
1
customerID
DOB
gender
1
generates
1
offerID
startDate
Customer
Offers
endDate
contact_no
1
place an
N
pay_date_time
password
discountPercentage
address
payment_mode
city
status
Invoice_id
Invoice
street
state
Total_amount
M
add to cart
N
quantity

Relation Schema Diagram:



Admin Contact	
adminID	varchar(4)
contact no	numeric(10)

Admin	
adminID	varchar(4)
name	varchar(20)
email	varchar(254)
username	varchar(30)
password	varchar(30)
status	boolean

Branch	
branchID	varchar(4)
name	varchar(20)
street	varchar(50)
city	varchar(50)
state	varchar(50)
startTime	varchar(10)
closedTime	varchar(10)
adminID	varchar(4)
distributorID	varchar(4)
openDate	Date
status	boolean

Batch	
batchNo	varchar(6)
arrival date time	timestamp

Batch_Receives	
branchID	varchar(4)
batchNo	varchar(6)

Distributor	
distrubutorID	varchar(4)
name	varchar(20)
usename	varchar(30)
email	varchar(254)
password	varchar(30)
street	varchar(50)
city	varchar(50)
state	varchar(50)
adminID	varchar(4)
status	boolean

Distributor Contact	
distributorID	varchar(4)
contact no	numeric(10)

Batch_Stock_Detail	
stockID	varchar(4)
qty_arrived	integer
qty_left	integer
period_scale	varchar(20)
best_before_period	varchar(20)
batchNo	varchar(6)

Feedback	
branchID	varchar(4)
customerID	varchar(4)
rating	integer
description	varchar(50)

invoice	
invoiceID	varchar(5)
orderID	varchar(4)
payment_date_time	timestamp
payment_mode	varchar(20)
total amt	decimal(10,2)

Customer	
customerID	varchar(4)
name	varchar(20)
email	varchar(254)
password	varchar(30)
street	varchar(50)
city	varchar(50)
state	varchar(50)
adminID	varchar(4)
gender	varchar(6)
DOB	date
status	boolean

Order	
orderID	varchar(4)
no_of_product	integer
delivery_date	timestamp
actual_delivery_date	timestamp
branchID	varchar(4)
customerID	varchar(4)
status	varchar(10)

Customer Contact	
customerID	varchar(4)
contact no	numeric(10)

Cart	
customerID	varchar(4)
productID	varchar(4)
qty	integer

Product	
productID	varchar(4)
name	varchar(20)
fssai	varchar(15)
price	decimal(10,2)
description	varchar(50)
scaleID	varhcar(4)
weigth/piece-per_packet	integer
offerID	varchar(5)
updated_price	decimal(10,2)
status	boolean

Stock_Contains	
stockID	varchar(4)
productID	varchar(4)

Order_includes	
orderID	varchar(4)
productID	varchar(4)
qty	integer

Scale	
scaleID	varchar(4)
scaleUnit	varchar(10)

Offer	
offerID	varchar(5)
discount_percentage	decimal(10,2)
startDate	timestamp
endDate	timestamp

FUNCTIONAL DEPENDENCIES AND CONTRAINTS CLASSIFICATION

Table : Admin

Table Attributes :

adminID, email, username, password, status

Functional Dependencies :

adminID -> email

adminID -> name

adminID -> username

adminID -> password

adminID -> status

email -> adminID

email -> name

email -> username

email -> password

email -> status

Constraints :

a) Primary Key : adminID
b) Foreign Key : None
c) Referential : Distributor Table, Admin_Contact Table, Branch Table, Customer Table
d) Domain :

```
adminID  varchar(4) PRIMARY KEY,
name  varchar(20) (NOT NULL),
username  varchar(30) (NOT NULL),
email  varchar(254) (NOT NULL) check (email LIKE '%@%.%'
AND email NOT LIKE '@%' AND email NOT LIKE '%@%@%'),
password  varchar(30)  (NOT NULL),
status boolean DEFAULT true
```

- Here, Not in BCNF Form because email determines other attributes also but email can not be assign as primary key because in scenario where email ID is blocked we can still access that tuple with adminID.

Table : Admin_Contact

Table Attributes :

adminID, contact_no

Functional Dependencies :

No Functional Dependency.

Constraints :

a) Primary Key : adminID, contact_no
b) Foreign Key : adminID
c) Referential : None
d) Domain :

```
adminID  varchar(4) PRIMARY KEY ,
contact_no  numeric(10)  PRIMARY KEY
```

- Here, our relation is in BCNF Form.

Table : Branch

Table Attributes :

branchID, name, street, city, state, startTime, closedTime, distributorID, openDate, status

Functional Dependencies :

branchID -> name

branchID -> status

branchID -> street

branchID -> city

branchID -> state

branchID -> startTime

branchID -> closedTime

branchID -> distributorID

branchID -> openDate

Constraints :

a) Primary Key : branchID
b) Foreign Key : distributorID, adminID
c) Referential : Branch_Receives Table, Feedback Table, Order Table

d) Domain :

```
branchID  varchar(4) PRIMARY KEY
name  varchar(20)
street  varchar(50)
city  varchar(50)
state  varchar(50)  (NOT NULL)
startTime  varchar(10)
closedTime  varchar(10)
adminID  varchar(4)   (NOT NULL)
distributorID  varchar(4)   (NOT NULL)
openDate  date
status boolean DEFAULT true
```

- Here, as we have branchID as key which defines all the attributes, so our relation is in BCNF Form.

Table : Distributor

Table Attributes :

distributorID, name, email, password, street, city, state, adminID, status

Functional Dependencies :

distributorID -> name

distributorID -> email

distributorID -> password

distributorID -> street

distributorID -> city

distributorID -> state

distributorID -> adminID

distributorID -> status

email -> distributorID

email -> name

email -> password

email -> street

email -> city

email -> state

email -> adminID

email -> status

Constraints :

a) Primary Key : distributorID
b) Foreign Key : adminID
c) Referential : Branch Tabel, Distributor_Contact Table
d) Domain :

```
distributorID  varchar(4) PRIMARY KEY
name  varchar(20)
username  varchar(30) (NOT NULL)
email  varchar(254)  (NOT NULL) check (email LIKE '%@%.%'
AND email NOT LIKE '@%' AND email NOT LIKE '%@%@%')
password  varchar(30) (NOT NULL)
street  varchar(50)
city  varchar(50)
state  varchar(50) (NOT NULL)
adminID  varchar(4) (NOT NULL)
status boolean DEFAULT true
```

- Here, Not in BCNF Form because email determines other attributes also but email can not be assign as primary key because in scenario where email ID is blocked we can still access that tuple with distributorID.

Table : Distributor_Contact

Table Attributes :

distributorID, contact_no

Functional Dependencies :

No Functional Dependency.

Constraints :

a) Primary Key : distributorID, contact_no
b) Foreign Key : distributorID
c) Referential : None
d) Domain :

```
distributorID  varchar(4) PRIMARY KEY
contact_no  numeric(10) PRIMARY KEY
```

- Here, our relation is in BCNF Form.

Table : Order

Table Attributes :

orderID, orderDate, no_of_product, deliveryDate, actual_deliver_date, branchID, customerID, status

Functional Dependencies :

orderID -> orderDate

orderID -> no_of_product

orderID -> deliveryDate

orderID -> actual_delivery_date

orderID -> status

orderID -> branchID

orderID -> customerID

Constraints :

a) Primary Key : orderID
b) Foreign Key : branchID, customerID
c) Referential : Invoice Table, Order_Includes Table
d) Domain :

```
orderID  varchar(4)
no_of_product  integer  (NOT NULL)
deliveryDate  timestamp  (NOT NULL)
actual_delivery_date  timestamp
branchID  varchar(4)  (NOT NULL)
customerID  varchar(4)  (NOT NULL)
status varchar(10) DEFAULT pending
```

- Here, as we have orderID as key which defines all the attributes, so our relation is in BCNF Form.

Table : Batch

Table Attributes :

batchNo, arrival_date_time

Functional Dependencies :

batchNo -> arrival_date_time

Constraints :

a) Primary Key : batchNo
b) Foreign Key : None
c) Referential : Batch_Receives Table, Batch_Stock_Details Table
d) Domain :

```
batchNo  varchar(6)
arrival_date_time  timestamp (NOT NULL)
```

- Here, as we have batchNo as key which defines all the attributes, so our relation is in BCNF Form.

Table : Batch_Receives

Table Attributes :

branchID, batchNo

Functional Dependencies :

No functional dependencies.

Constraints :

a) Primary Key : branchID, batchNo
b) Foreign Key : branchID, batchNo
c) Referential : None
d) Domain :

```
branchID  varchar(4) PRIMARY KEY
batchNo  varchar(6)  PRIMARY KEY
```

- Here, our relation is in BCNF Form .

Table : Feedback

Table Attributes :

branchId, customerID, rating, description

Functional Dependencies :

branchID, customerID -> rating

branchID, customerID -> description

Constraints :

a) Primary Key : branchID, customerID
b) Foreign Key : branchID, customerID
c) Referential : None

d) Domain :

```
branchID  varchar(4) PRIMARY KEY
customerID  varchar(4) PRIMARY KEY
rating  integer  (NOT NULL)
description  varchar(50)
```

- Here, as we have branchID and customerID as key which defines all the attributes, so our relation is in BCNF Form.

Table : Customer

Table Attributes :

customerID, name , email, password, street, city, state, adminID, gender, DOB, status

Functional Dependencies :

customerID -> name

customerID -> email

customerID -> password

customerID -> street

customerID -> city

customerID -> state

customerID -> adminID

customerID -> gender

customerID -> DOB

customerID -> status

email -> customerID

email -> name

email -> password

email -> street

email -> city

email -> state

email -> adminID

email -> gender

email -> DOB

email -> status

Constraints :

a) Primary Key : customerID
b) Foreign Key : adminID
c) Referential: Customer_Contact Table, Order Table, Cart Table
d) Domain :

```
customerID  varchar(4)  PRIMARY KEY
name  varchar(20)
email  varchar(254) (NOT NULL) check (email LIKE '%@%.%'
AND email NOT LIKE '@%' AND email NOT LIKE '%@%@%')
password  varchar(30) (NOT NULL)
street  varchar(50)
city  varchar(50)
state  varchar(50)
adminID  varchar(4) (NOT NULL)
gender varchar(6) CHECK(gender in('male','female'))
DOB  date
status boolean DEFAULT true
```

- Here, Not in BCNF Form because email determines other attributes also but email can not be assign as primary key because in scenario where email ID is blocked we can still access that tuple with customerID.

Table : Customer_Contact

Table Attributes :

customerID, contact_no

Functional Dependencies :

No Functional Dependency.

Constraints :

a) Primary Key : customerID, contact_no
b) Foreign Key : customerID
c) Referential : None
d) Domain :

```
customerID varchar(4)  PRIMARY KEY
contact_no numeric(10)
```

- Here, our relation is in BCNF Form.

Table : Cart

Table Attributes :

customerID, productid, qty

Functional Dependencies :

customerID, productID -> qty

Constraints :

a) Primary Key : customerID, productID
b) Foreign Key : customerID, productid
c) Referential : None
d) Domain :

```
customerID  varchar(4)  PRIMARY KEY
productID  varchar(4)  PRIMARY KEY
qty  integer (NOT NULL)
```

- Here, as we have customerID and productID as key which defines all the attributes, so our relation is in BCNF Form.

Table : Scale

Table Attributes :

scaleID, scaleUnit

Functional Dependencies :

scaleID -> scaleUnit

Constraints :

a) Primary Key : scaleID
b) Foreign Key : None
c) Referential : Product Table
d) Domain :

```
scaleID  varchar(4)  PRIMARY KEY
scaleUnit  varchar(10)  (NOT NULL)
```

- Here, as we have scaleID as key which defines all the attributes, so our relation is in BCNF Form.

Table : Product

Table Attributes :

prodID, name, fssai, price, description, status, scaleID, weight/piece per packet, offerID, updated_price

Functional Dependencies :

prodID -> name

prodID -> fssai

prodID -> price

prodID -> description

prodID -> status

prodID -> scaleID

prodID -> weight/piece per packet

prodID -> offerID

prodID- > updated_price

prodID -> name

fssai -> prodID

fssai -> price

fssai -> description

fssai -> status

fssai -> scaleID

fssai -> weight/piece per packet

fssai -> offerID

fssai - > updated_price

Constraints :

a) Primary Key : prodID
b) Foreign Key : scaleID, offerID
c) Referential : Cart Table, Stock_Contains Table, Order_Includes Table
d) Domain :

```
prodID  varchar(4) PRIMARY KEY
name  varchar(20) (NOT NULL)
fssai  varchar(15) (NOT NULL)
price  decimal(10,2)(NOT NULL)
description  varchar(50)
scaleID  varchar(4)(NOT NULL)
weight/piece_per_packet  integer (NOT NULL)
offerID  varchar(5)
updated_price  decimal(10,2)
status oolean DEFAULT true
```

- Here, Not in BCNF Form because fssai which is a non-prime key attribute determines other attributes also. But for simplicity we are using prodID as primary key.

Table : Order_Includes

Table Attributes :

prodID, orderID, qty

Functional Dependencies :

orderID, prodID -> qty

Constraints :

a) Primary Key : orderID, prodID
b) Foreign Key : orderID, prodID
c) Referential : None
d) Domain :

```
orderID  varchar(4) PRIMARY KEY
prodID  varchar(4)  PRIMARY KEY
qty  integer (NOT NULL)
```

- Here, as we have orderID and prodID as key which defines all the attributes, so our relation is in BCNF Form.

Table : Batch_Stock_Detail

Table Attributes :

stockID, qty_arrive, qty_left, period_scale, best_before_period, batchNo

Functional Dependencies :

stockID -> qty_arrive

stockID -> qty_left

stockID -> period_scale

stockID -> best_before_period

stockID -> batchNo

Constraints :

a) Primary Key : stockID
b) Foreign Key : batchNo
c) Referential : Stock_Contains Table
d) Domain :

```
stockID  varchar(4)  PRIMARY KEY
qty_arrive  integer (NOT NULL)
qty_left  integer (NOT NULL)
period_scale  varchar(20)
best_before_period  varchar(20)
batchNo  varchar(6) (NOT NULL)
```

- Here, as we have stockID as key which defines all the attributes, so our relation is in BCNF Form.

Table : Stock_Contains

Table Attributes :

stockID, prodID

Functional Dependencies :

No Functional Dependencies.

Constraints :

a) Primary Key : stockID, prodID
b) Foreign Key : stockID, prodID
c) Referential : None
d) Domain :

```
stockID  varchar(4)  PRIMARY KEY
prodID  varchar(4)  PRIMARY KEY
```

- Here, our relation is in BCNF Form

Table : Invoice

Table Attributes :

invoiceID, orderID, payment_date_time, payment_mode, total_amt

Functional Dependencies :

invoiceID -> orderID

invoiceID -> payment_date_time

invoiceID -> payment_mode

invoiceID -> total_amt

Constraints :

a) Primary Key : invoiceID
b) Foreign Key : None
c) Referential : None

d) Domain :

```
invoiceID  varchar(5)  PRIMARY KEY
orderID  varchar(4) (NOT NULL)
payment_date_time  timestamp (NOT NULL)
payment_mode  varchar(20) (NOT NULL)
total_amt  decimal(10,2) (NOT NULL)
```

- Here, as we have invoiceID as key which defines all the attributes, so our relation is in BCNF Form.

Table : Offer

Table Attributes :

offerID, discount_percentage, startDate, endDate

Functional Dependencies :

offerID -> discount_percentage

offerID -> startDate

offerID -> endDate

Constraints :

a) Primary Key : offerID
b) Foreign Key : none
c) Referential :
d) Domain :

```
offerID  varchar(5) PRIMARY KEY
discount_percentage  integer (NOT NULL)
startDate  timestamp  (NOT NULL)
endDate  timestamp  (NOT NULL)
```

- Here, as we have offerID as key which defines all the attributes, so our relation is in BCNF Form.

Trigger and function:

1. Price update by offer

```
create or replace function price_update()
    returns trigger as $$
    DECLARE
    perccent numeric;
    begin
    set search_path to gms14;
        select discount_percentage into perccent from offer where offerid = 
New.offerid;
        select  (New.price-((New.price * perccent)/100)) into perccent;
        if (TG_OP = 'INSERT') THEN
            raise notice 'bf cond';
            IF (NEW.offerid is NULL And New.updated_price is not NULL) 
THEN
            raise notice 'af cond';
                update product set updated_price = NULL where 
prodid=NEW.prodid;
            ELSEIF (new.updated_price!=perccent OR (NEW.offerid is not 
NULL and New.updated_price is NULL)) THEN
                update product set updated_price=(New.price-(((select 
discount_percentage from offer where offerid = New.offerid) * New.price)/100)) 
where prodid=NEW.prodid;
            END IF;
        ELSEIF (TG_OP = 'UPDATE') THEN
            raise notice 'bf cond';
            IF (NEW.offerid is NULL And New.updated_price is not NULL) 
THEN
            raise notice 'af cond';
                update product set updated_price=NULL where 
prodid=NEW.prodid;
```

```
            ELSEIF (new.updated_price!=perccent OR (NEW.offerid is not NULL and New.updated_price is NULL)) THEN

            raise notice 'af cond';

                update product set updated_price=(New.price-(((select discount_percentage from offer where offerid = New.offerid) * New.price)/100)) where prodid=NEW.prodid;

            END IF;

        END IF;


    RETURN New;

    END;

$$ LANGUAGE plpgsql;

CREATE TRIGGER UP_PRICE

AFTER INSERT OR UPDATE ON product

    FOR EACH ROW EXECUTE PROCEDURE price_update();
```

2. Count total amount for invoice

```
set search_path to gms14;
create or replace function count_ttlamount()
    returns trigger as $$
    DECLARE
        ttlprice integer;
    begin
    set search_path to gms14;

        select sum((CASE WHEN updated_price is NULL then price else
updated_price end)  * qty) into ttlprice from product p JOIN order_includes o
ON(o.prodid=p.prodid) join tblorder t on (o.orderid=t.orderid) where t.orderid =
new.orderid  group by t.orderid  order by t.orderid;
        new.total_amt := ttlprice;
    return new;
    END;
$$ LANGUAGE plpgsql;
CREATE TRIGGER count_ttlamount
before INSERT or UPDATE ON invoice
    FOR EACH ROW EXECUTE PROCEDURE count_ttlamount();
```

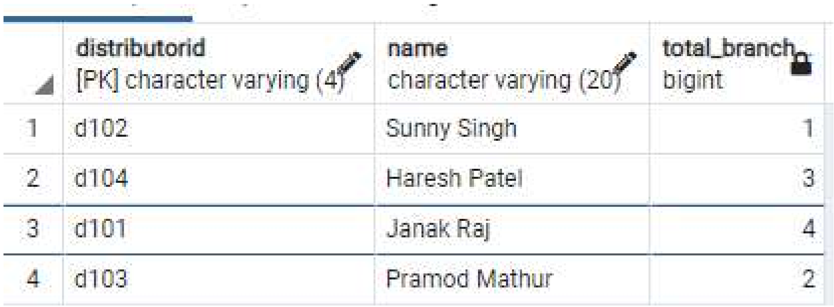
3. Remove offer after expire

```
set search_path to gms14;
create or replace function updateoffer()
    return integer
    as $$
    begin
    update product set offerid=null where offerid in (select offerid from offer
where CURRENT_DATE not between startdate and enddate );
    END;
$$ LANGUAGE plpgsql;
```

QUERIES:

1. Display the total number of branches handled by all the distributors.

```
select d.distributorid, d.name, count(branchid) as total_branch

from branch b

JOIN distributor d

ON(d.distributorid=b.distributorid)

group by d.distributorid;
```



	distributorid [PK] character varying (4)	name character varying (20)	total_branch bigint
1	d102	Sunny Singh	1
2	d104	Haresh Patel	3
3	d101	Janak Raj	4
4	d103	Pramod Mathur	2

2. Display the detail of branch which receive rating less than 6 more than 2 times.

select b.*, count(b.branchid)

from branch b

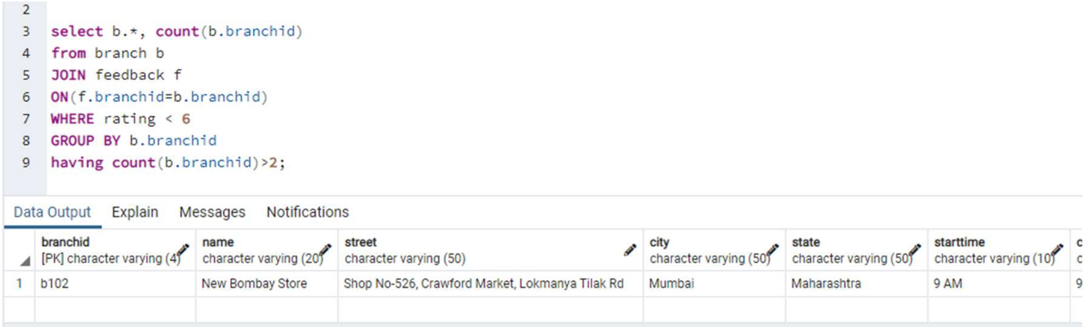
JOIN feedback f

ON(f.branchid=b.branchid)

WHERE rating < 6

GROUP BY b.branchid

having count(b.branchid)>2;

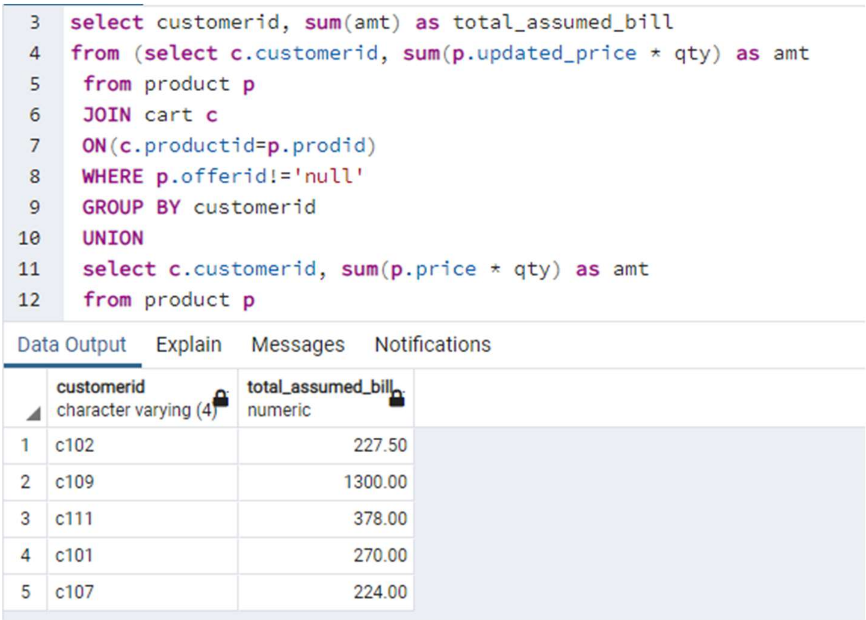


```

select b.*, count(b.branchid)
from branch b
JOIN feedback f
ON(f.branchid=b.branchid)
WHERE rating < 6
GROUP BY b.branchid
having count(b.branchid)>2;
```

	branchid [PK] character varying (4)	name character varying (20)	street character varying (50)	city character varying (50)	state character varying (50)	starttime character varying (10)	c c
1	b102	New Bombay Store	Shop No-526, Crawford Market, Lokmanya Tilak Rd	Mumbai	Maharashtra	9 AM	9

3. Calculate the total price of all the products saved in the cart of all the customers.

```
select customerid, sum(amt) as total_assumed_bill
from (select c.customerid, sum(p.updated_price * qty) as amt
        from product p
        JOIN cart c
        ON(c.productid=p.prodid)
        WHERE p.offerid!='null'
        GROUP BY customerid
        UNION
        select c.customerid, sum(p.price * qty) as amt
        from product p
        JOIN cart c
        ON(c.productid=p.prodid)
        WHERE p.offerid is NULL
        GROUP BY customerid) r
GROUP BY customerid;
```



```
select customerid, sum(amt) as total_assumed_bill
from (select c.customerid, sum(p.updated_price * qty) as amt
 from product p
 JOIN cart c
 ON(c.productid=p.prodid)
 WHERE p.offerid!='null'
 GROUP BY customerid
 UNION
 select c.customerid, sum(p.price * qty) as amt
 from product p
```

	customerid character varying (4)	total_assumed_bill numeric
1	c102	227.50
2	c109	1300.00
3	c111	378.00
4	c101	270.00
5	c107	224.00

4. List product details of top 3 most sold product based on quantity.

select p.* from product p

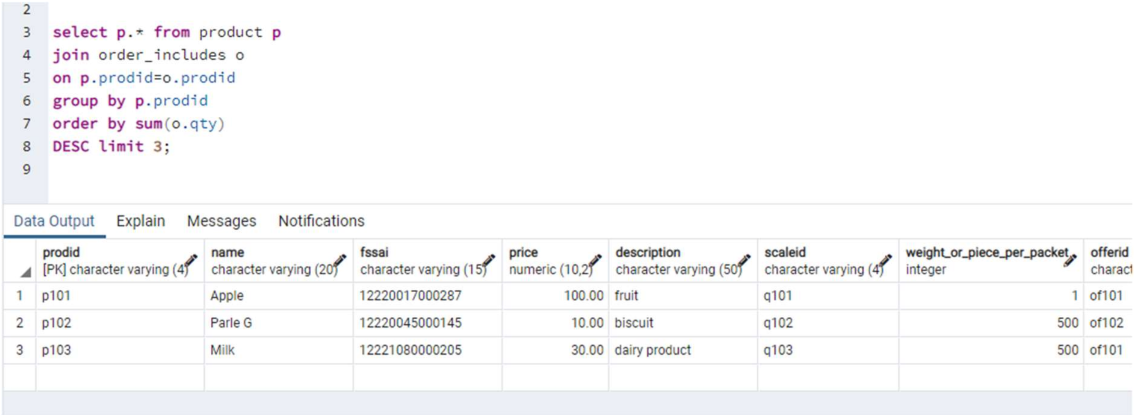
join order_includes o

on p.prodid=o.prodid

group by p.prodid

order by sum(o.qty)

DESC limit 3;



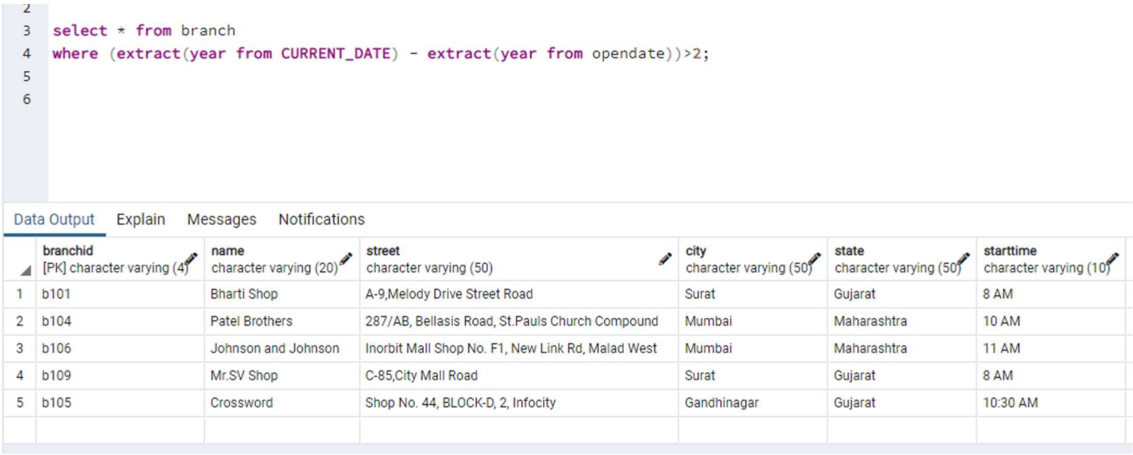
```
select p.* from product p
join order_includes o
on p.prodid=o.prodid
group by p.prodid
order by sum(o.qty)
DESC limit 3;
```

	prodid [PK] character varying (4)	name character varying (20)	fssai character varying (15)	price numeric (10,2)	description character varying (50)	scaleid character varying (4)	weight_or_piece_per_packet integer	offerid charact
1	p101	Apple	12220017000287	100.00	fruit	q101	1	of101
2	p102	Parle G	12220045000145	10.00	biscuit	q102	500	of102
3	p103	Milk	12221080000205	30.00	dairy product	q103	500	of101

5. Display the detail of all the branch which opened before 2 year.

select * from branch

where (extract(year from CURRENT_DATE) - extract(year from opendate))>2;

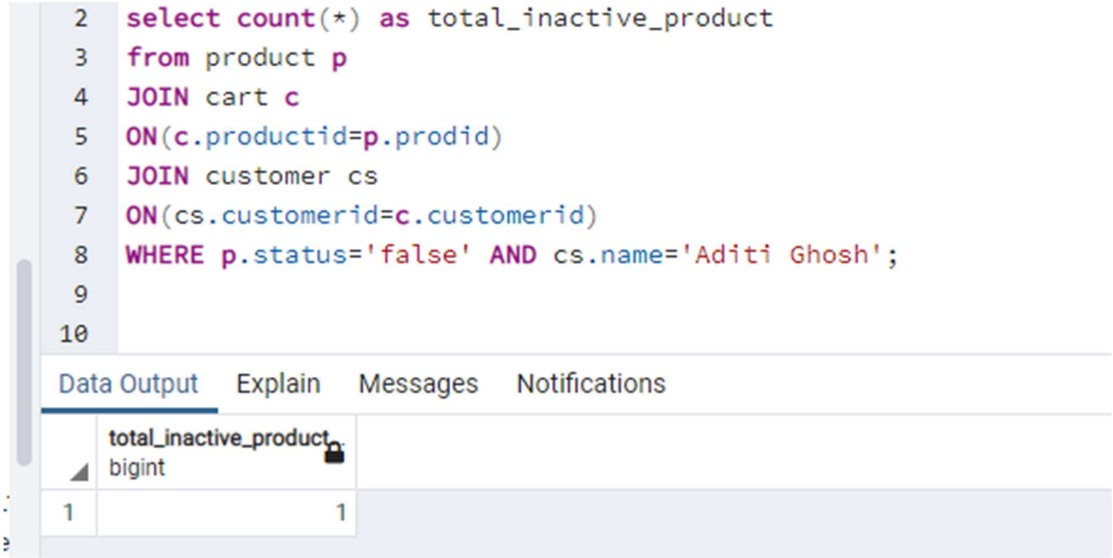


```
select * from branch
where (extract(year from CURRENT_DATE) - extract(year from opendate))>2;
```

	branchid [PK] character varying (4)	name character varying (20)	street character varying (50)	city character varying (50)	state character varying (50)	starttime character varying (10)
1	b101	Bharti Shop	A-9,Melody Drive Street Road	Surat	Gujarat	8 AM
2	b104	Patel Brothers	287/AB, Bellasis Road, St.Pauls Church Compound	Mumbai	Maharashtra	10 AM
3	b106	Johnson and Johnson	Inorbit Mall Shop No. F1, New Link Rd, Malad West	Mumbai	Maharashtra	11 AM
4	b109	Mr.SV Shop	C-85,City Mall Road	Surat	Gujarat	8 AM
5	b105	Crossword	Shop No. 44, BLOCK-D, 2, Infocity	Gandhinagar	Gujarat	10:30 AM

6. Count the product saved in the cart of customer named ‘Aditi Ghosh’ which are no longer active.

```
select count(*) as total_inactive_product

from product p

JOIN cart c

ON(c.productid=p.prodid)

JOIN customer cs

ON(cs.customerid=c.customerid)

WHERE p.status='false' AND cs.name='Aditi Ghosh';
```

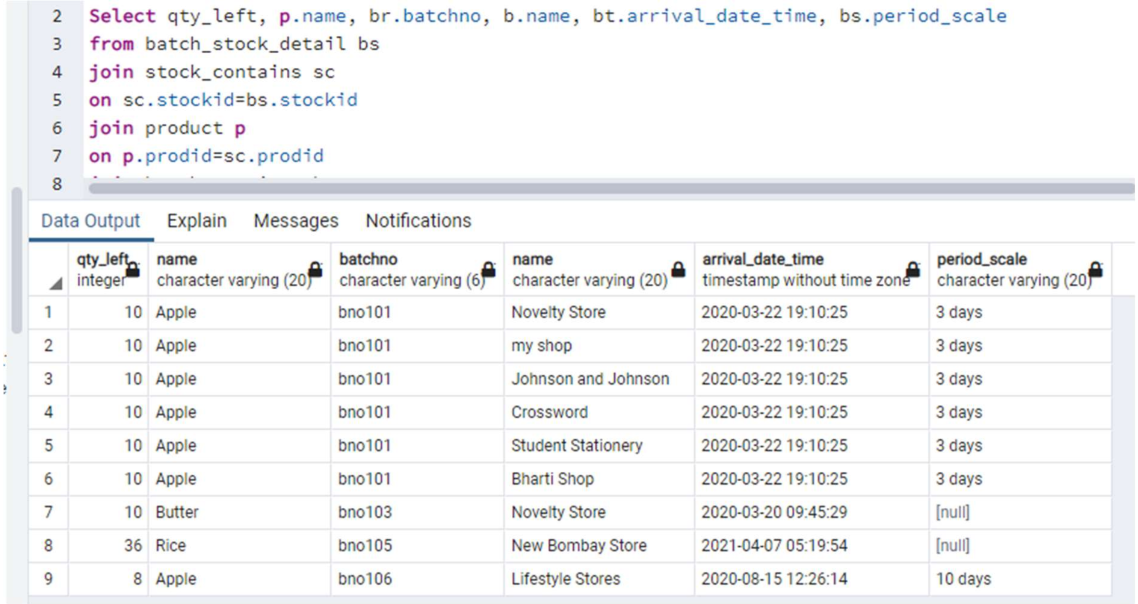


```
select count(*) as total_inactive_product
from product p
JOIN cart c
ON(c.productid=p.prodid)
JOIN customer cs
ON(cs.customerid=c.customerid)
WHERE p.status='false' AND cs.name='Aditi Ghosh';
```

	total_inactive_product bigint
1	1

7. Count product for all the offers.

```
Select qty_left, p.name, br.batchno, b.name, bt.arrival_date_time, bs.period_scale

from batch_stock_detail bs

join stock_contains sc

on sc.stockid=bs.stockid

join product p

on p.prodid=sc.prodid

join batch_receives br

on bs.batchno=br.batchno

join branch b

on b.branchid=br.branchid

join batch bt

on bt.batchno=br.batchno

where (period_scale is null

        and (to_date(concat('01',' ',split_part(best_before_period,' ',1),' ',split_part(best_before_period,' ',2)),'DD mon yyyy')<current_date))

        or cast(split_part(cast((current_date-bt.arrival_date_time) as varchar),' ',1)as integer)>cast(split_part(period_scale,' ',1)as integer);
```



```
Select qty_left, p.name, br.batchno, b.name, bt.arrival_date_time, bs.period_scale
from batch_stock_detail bs
join stock_contains sc
on sc.stockid=bs.stockid
join product p
on p.prodid=sc.prodid
```

	qty_left integer	name character varying (20)	batchno character varying (6)	name character varying (20)	arrival_date_time timestamp without time zone	period_scale character varying (20)
1	10	Apple	bno101	Novelty Store	2020-03-22 19:10:25	3 days
2	10	Apple	bno101	my shop	2020-03-22 19:10:25	3 days
3	10	Apple	bno101	Johnson and Johnson	2020-03-22 19:10:25	3 days
4	10	Apple	bno101	Crossword	2020-03-22 19:10:25	3 days
5	10	Apple	bno101	Student Stationery	2020-03-22 19:10:25	3 days
6	10	Apple	bno101	Bharti Shop	2020-03-22 19:10:25	3 days
7	10	Butter	bno103	Novelty Store	2020-03-20 09:45:29	[null]
8	36	Rice	bno105	New Bombay Store	2021-04-07 05:19:54	[null]
9	8	Apple	bno106	Lifestyle Stores	2020-08-15 12:26:14	10 days

8. Display the detail of customer who paid total amount more than average of all the invoice total amount.

select c.*

from customer c

JOIN tblorder o

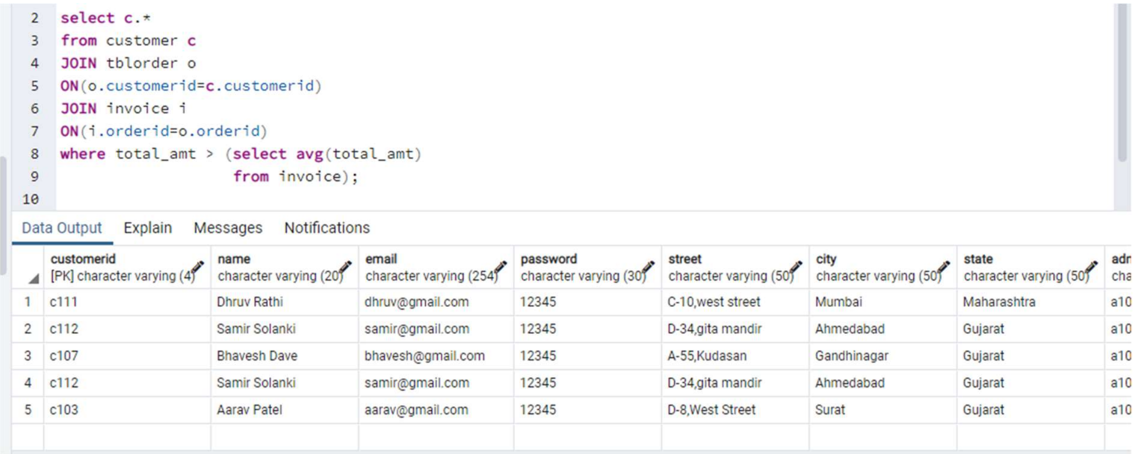
ON(o.customerid=c.customerid)

JOIN invoice i

ON(i.orderid=o.orderid)

where total_amt > (select avg(total_amt)

from invoice);



```
select c.*
from customer c
JOIN tblorder o
ON(o.customerid=c.customerid)
JOIN invoice i
ON(i.orderid=o.orderid)
where total_amt > (select avg(total_amt)
                   from invoice);
```

	customerid [PK] character varying (4)	name character varying (20)	email character varying (254)	password character varying (30)	street character varying (50)	city character varying (50)	state character varying (50)	adr cha
1	c111	Dhruv Rathi	dhruv@gmail.com	12345	C-10,west street	Mumbai	Maharashtra	a10
2	c112	Samir Solanki	samir@gmail.com	12345	D-34,gita mandir	Ahmedabad	Gujarat	a10
3	c107	Bhavesh Dave	bhavesh@gmail.com	12345	A-55,Kudasan	Gandhinagar	Gujarat	a10
4	c112	Samir Solanki	samir@gmail.com	12345	D-34,gita mandir	Ahmedabad	Gujarat	a10
5	c103	Aarav Patel	aarav@gmail.com	12345	D-8,West Street	Surat	Gujarat	a10

9. Count the no. of customer for all the product which were saved in their cart.

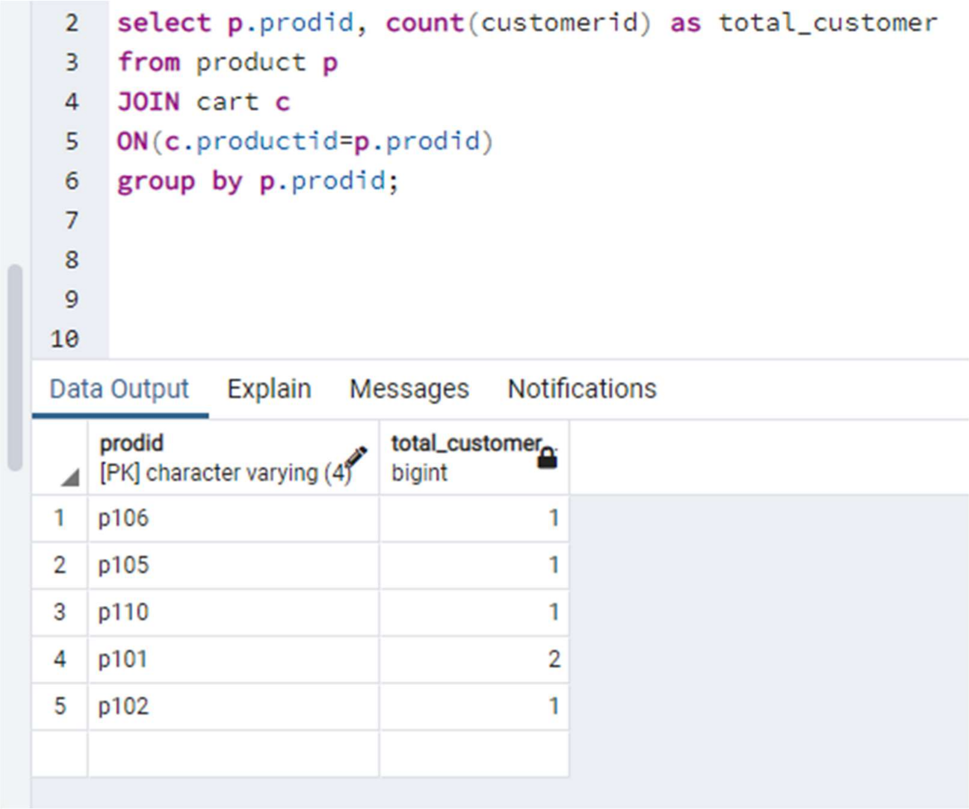
select p.prodid, count(customerid) as total_customer

from product p

JOIN cart c

ON(c.productid=p.prodid)

group by p.prodid;



```
select p.prodid, count(customerid) as total_customer
from product p
JOIN cart c
ON(c.productid=p.prodid)
group by p.prodid;
```

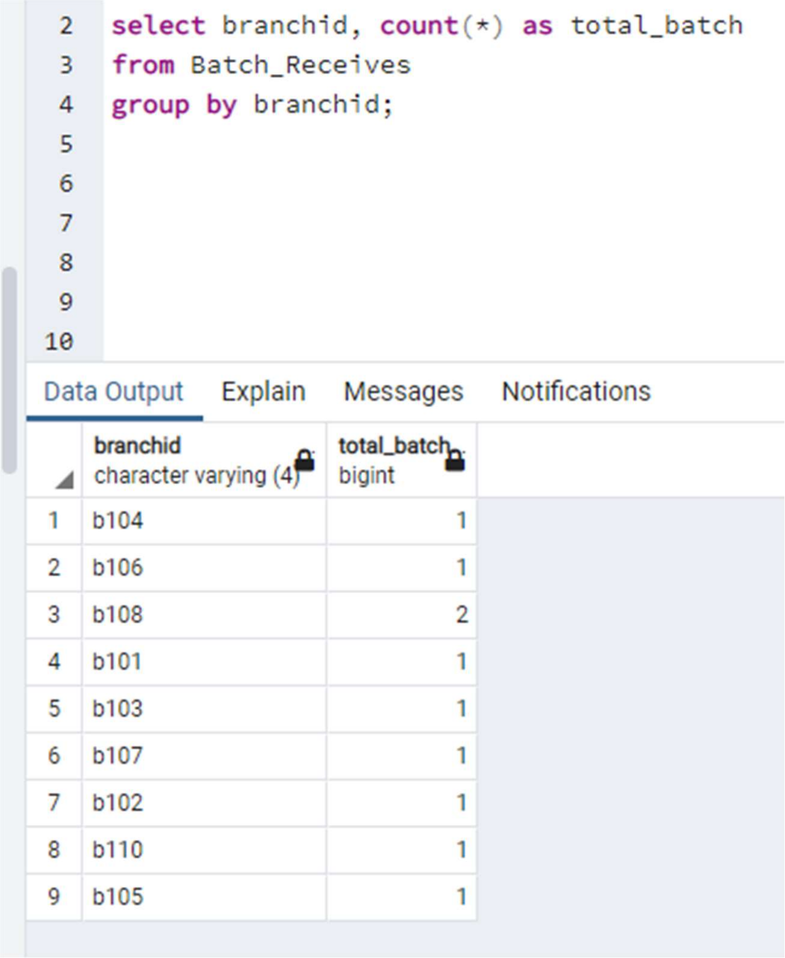
	prodid [PK] character varying (4)	total_customer bigint
1	p106	1
2	p105	1
3	p110	1
4	p101	2
5	p102	1

10. Count total number of batch received by branch.

select branchid, count(*) as total_batch

from Batch_Receives

group by branchid;



```
select branchid, count(*) as total_batch
from Batch_Receives
group by branchid;
```

	branchid character varying (4)	total_batch bigint
1	b104	1
2	b106	1
3	b108	2
4	b101	1
5	b103	1
6	b107	1
7	b102	1
8	b110	1
9	b105	1

11. Display the detail of branch which receive rating more than 7 where customer name is 'Akash Raval '.

select b.*, count(b.branchid)

from branch b

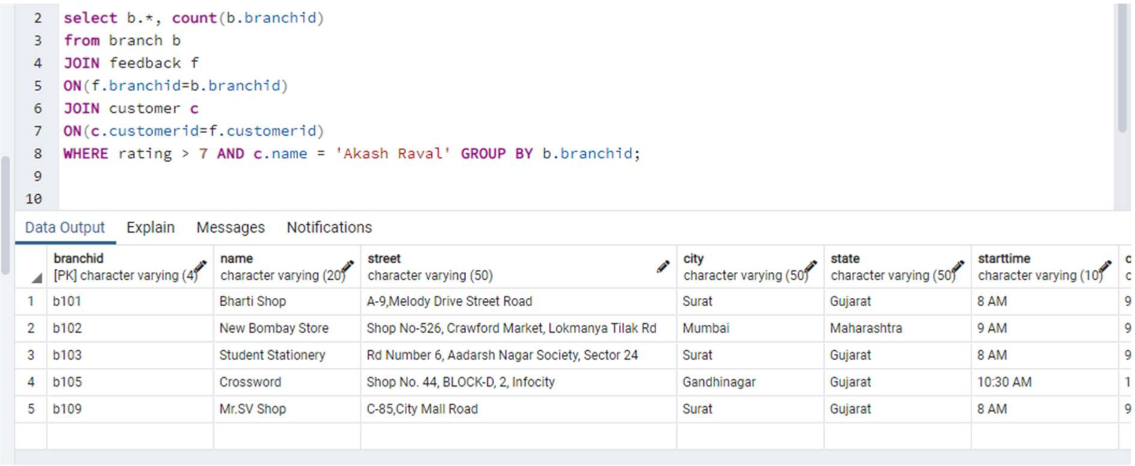
JOIN feedback f

ON(f.branchid=b.branchid)

JOIN customer c

ON(c.customerid=f.customerid)

WHERE rating > 7 AND c.name = 'Akash Raval' GROUP BY b.branchid;

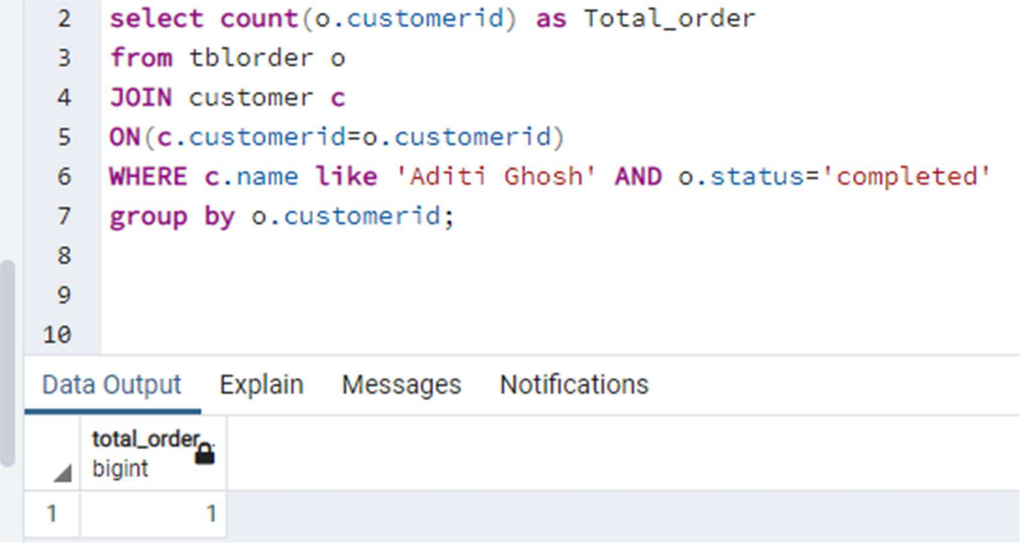


```
select b.*, count(b.branchid)
from branch b
JOIN feedback f
ON(f.branchid=b.branchid)
JOIN customer c
ON(c.customerid=f.customerid)
WHERE rating > 7 AND c.name = 'Akash Raval' GROUP BY b.branchid;
```

	branchid [PK] character varying (4)	name character varying (20)	street character varying (50)	city character varying (50)	state character varying (50)	starttime character varying (10)	c c
1	b101	Bharti Shop	A-9,Melody Drive Street Road	Surat	Gujarat	8 AM	9
2	b102	New Bombay Store	Shop No-526, Crawford Market, Lokmanya Tilak Rd	Mumbai	Maharashtra	9 AM	9
3	b103	Student Stationery	Rd Number 6, Aadarsh Nagar Society, Sector 24	Surat	Gujarat	8 AM	9
4	b105	Crossword	Shop No. 44, BLOCK-D, 2, Infocity	Gandhinagar	Gujarat	10:30 AM	1
5	b109	Mr.SV Shop	C-85,City Mall Road	Surat	Gujarat	8 AM	9

12. Total no. order placed by customer ID = c101 with status is completed.

```
select count(o.customerid) as Total_order
from tblorder o
JOIN customer c
ON(c.customerid=o.customerid)
WHERE c.name like 'Aditi Ghosh' AND o.status='completed'
group by o.customerid;
```



```
select count(o.customerid) as Total_order
from tblorder o
JOIN customer c
ON(c.customerid=o.customerid)
WHERE c.name like 'Aditi Ghosh' AND o.status='completed'
group by o.customerid;
```

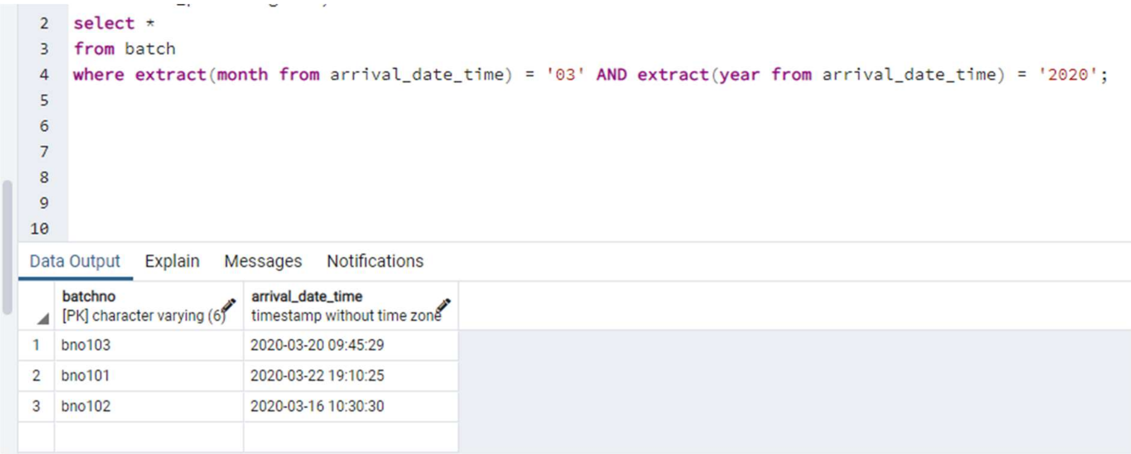
	total_order bigint
1	1

13. Display the details of all the batch which were received in March 2020.

select *

from batch

where extract(month from arrival_date_time) = '03' AND extract(year from arrival_date_time) = '2020';



```
select *
from batch
where extract(month from arrival_date_time) = '03' AND extract(year from arrival_date_time) = '2020';
```

	batchno [PK] character varying (6)	arrival_date_time timestamp without time zone
1	bno103	2020-03-20 09:45:29
2	bno101	2020-03-22 19:10:25
3	bno102	2020-03-16 10:30:30

14. Display all the detail of distributor who handled the branch which got rating = 4

select distinct d.*

from branch b

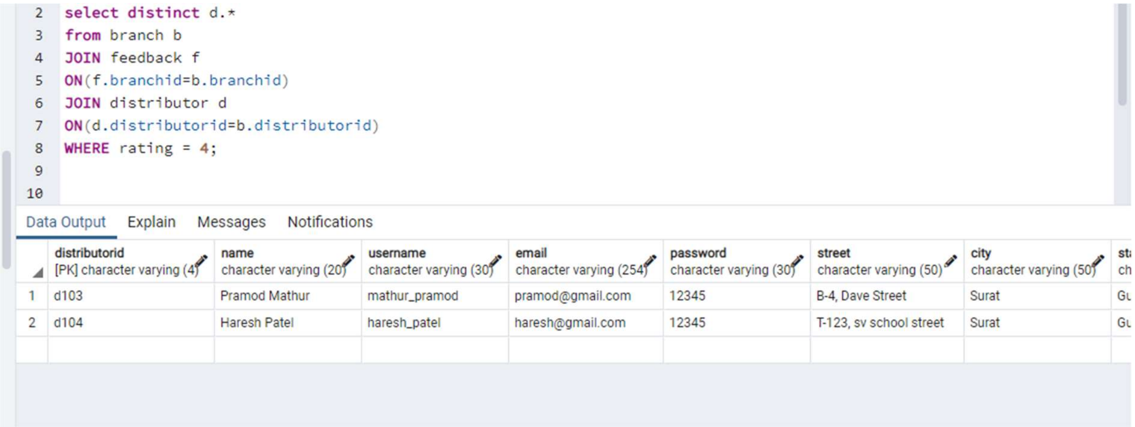
JOIN feedback f

ON(f.branchid=b.branchid)

JOIN distributor d

ON(d.distributorid=b.distributorid)

WHERE rating = 4;



```
select distinct d.*
from branch b
JOIN feedback f
ON(f.branchid=b.branchid)
JOIN distributor d
ON(d.distributorid=b.distributorid)
WHERE rating = 4;
```

	distributorid [PK] character varying (4)	name character varying (20)	username character varying (30)	email character varying (254)	password character varying (30)	street character varying (50)	city character varying (50)	st ch
1	d103	Pramod Mathur	mathur_pramod	pramod@gmail.com	12345	B-4, Dave Street	Surat	Gu
2	d104	Haresh Patel	haresh_patel	haresh@gmail.com	12345	T-123, sv school street	Surat	Gu

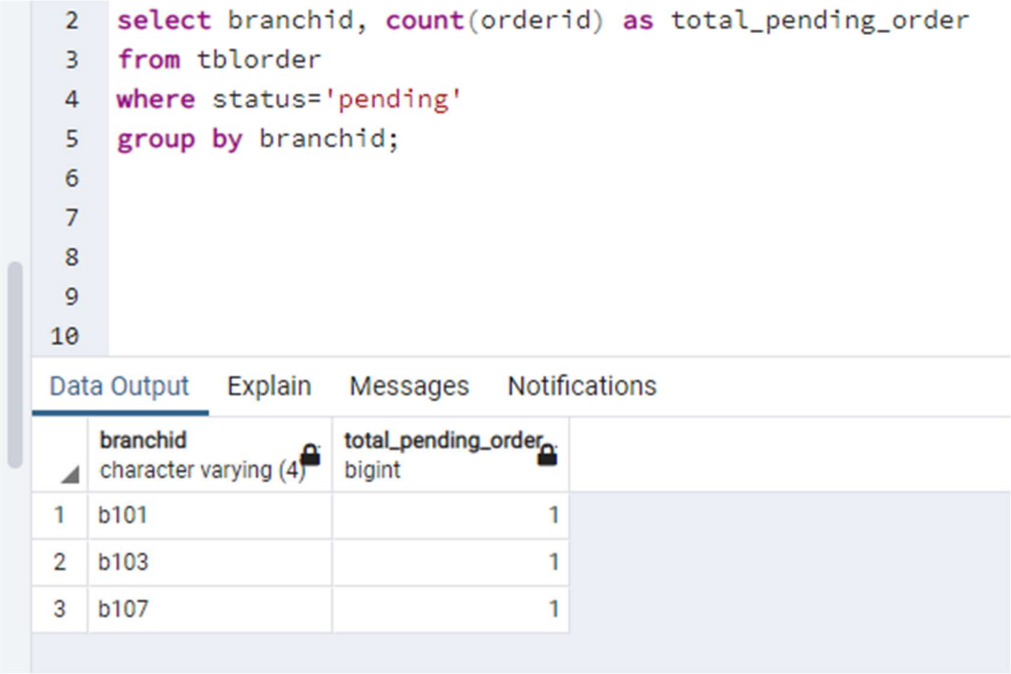
15. Count total order which are pending for branch.

select branchid, count(orderid) as total_pending_order

from tblorder

where status='pending'

group by branchid;



```
select branchid, count(orderid) as total_pending_order
from tblorder
where status='pending'
group by branchid;
```

	branchid character varying (4)	total_pending_order bigint
1	b101	1
2	b103	1
3	b107	1

16. Display the product detail saved in the cart of customerid ‘c102’ which has discount more than 8 percentage.

select p.*

from product p

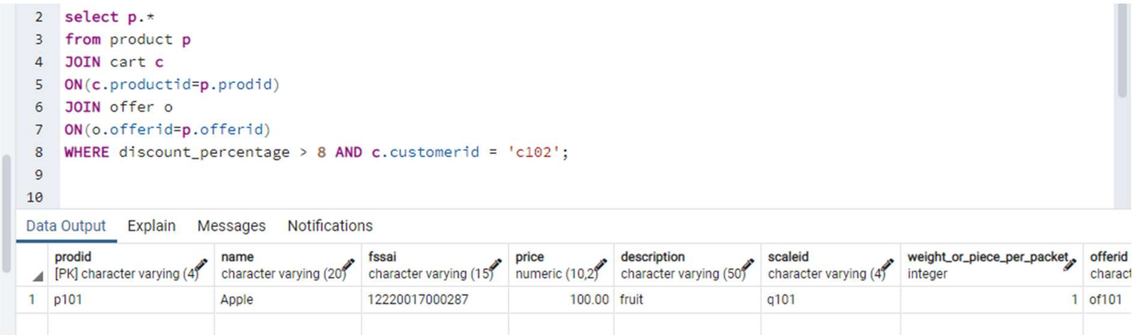
JOIN cart c

ON(c.productid=p.prodid)

JOIN offer o

ON(o.offerid=p.offerid)

WHERE discount_percentage > 8 AND c.customerid = 'c102';



```
select p.*
from product p
JOIN cart c
ON(c.productid=p.prodid)
JOIN offer o
ON(o.offerid=p.offerid)
WHERE discount_percentage > 8 AND c.customerid = 'c102';
```

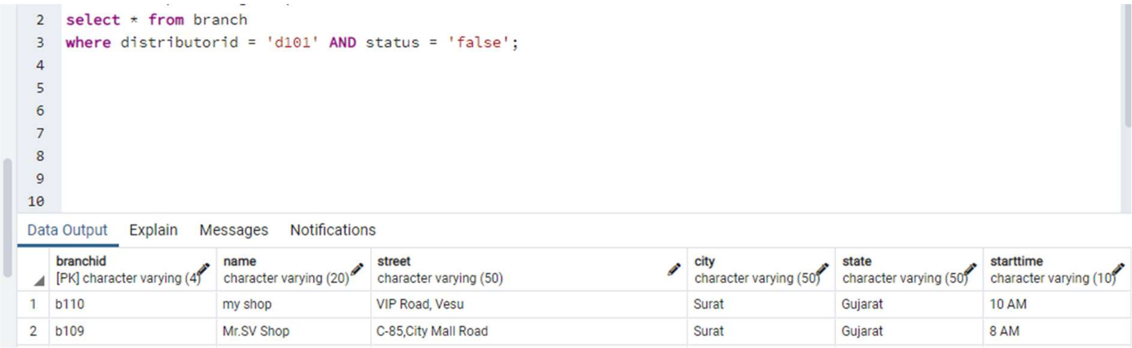
Data Output Explain Messages Notifications

	prodid [PK] character varying (4)	name character varying (20)	fssai character varying (15)	price numeric (10,2)	description character varying (50)	scaleid character varying (4)	weight_or_piece_per_packet integer	offerid charact
1	p101	Apple	12220017000287	100.00	fruit	q101	1	of101

17. Display all branch detail with status inactive handled by distributorid d101.

select * from branch

where distributorid = 'd101' AND status = 'false';



```
select * from branch
where distributorid = 'd101' AND status = 'false';
```

Data Output Explain Messages Notifications

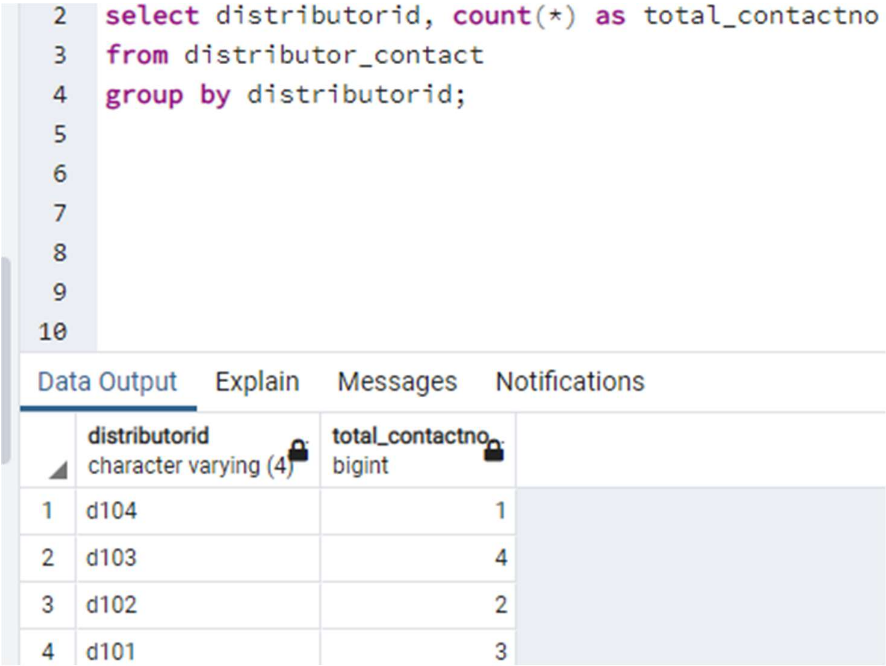
	branchid [PK] character varying (4)	name character varying (20)	street character varying (50)	city character varying (50)	state character varying (50)	starttime character varying (10)
1	b110	my shop	VIP Road, Vesu	Surat	Gujarat	10 AM
2	b109	Mr.SV Shop	C-85,City Mall Road	Surat	Gujarat	8 AM

18. Count total number of contact number of all the distributors.

select distributorid, count(*) as total_contactno

from distributor_contact

group by distributorid;



```
select distributorid, count(*) as total_contactno
from distributor_contact
group by distributorid;
```

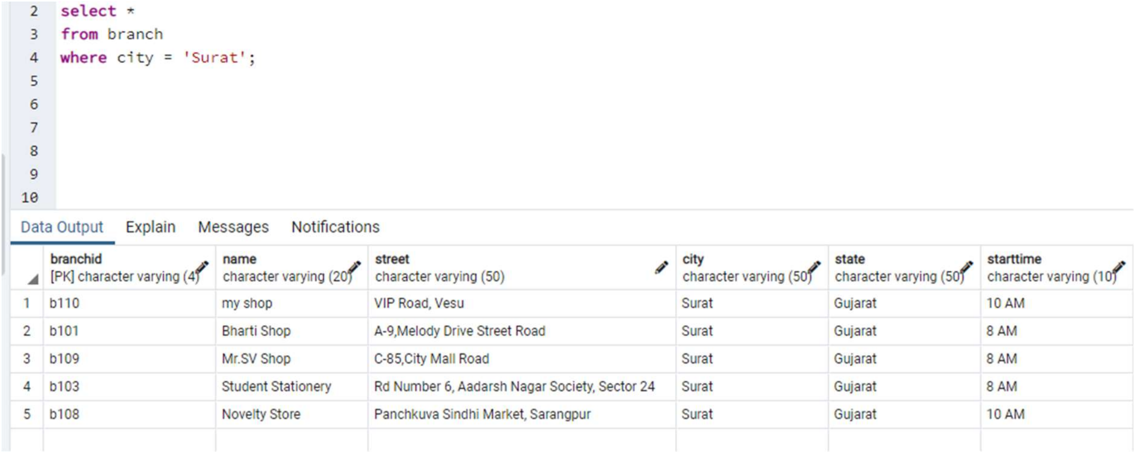
	distributorid character varying (4)	total_contactno bigint
1	d104	1
2	d103	4
3	d102	2
4	d101	3

19. Display the detail of all the branch in city Surat.

select *

from branch

where city = 'Surat';



```
select *
from branch
where city = 'Surat';
```

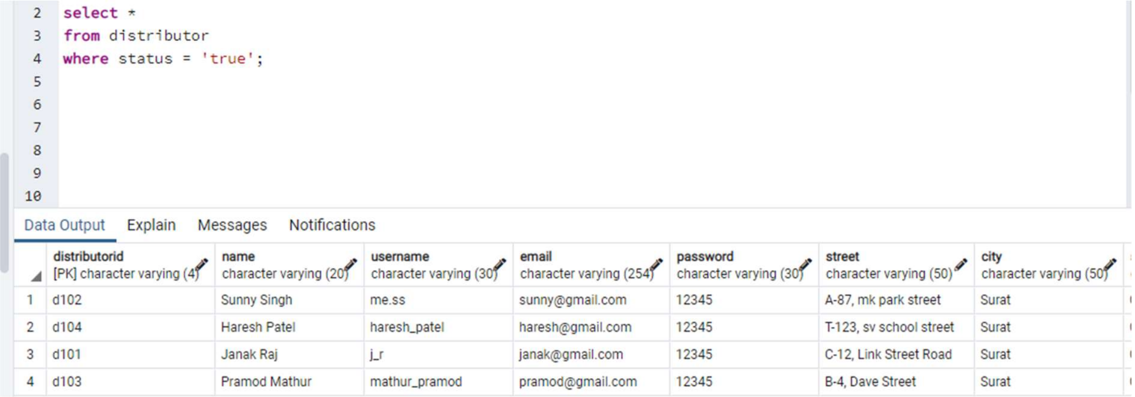
	branchid [PK] character varying (4)	name character varying (20)	street character varying (50)	city character varying (50)	state character varying (50)	starttime character varying (10)
1	b110	my shop	VIP Road, Vesu	Surat	Gujarat	10 AM
2	b101	Bharti Shop	A-9,Melody Drive Street Road	Surat	Gujarat	8 AM
3	b109	Mr.SV Shop	C-85,City Mall Road	Surat	Gujarat	8 AM
4	b103	Student Stationery	Rd Number 6, Aadarsh Nagar Society, Sector 24	Surat	Gujarat	8 AM
5	b108	Novelty Store	Panchkuva Sindhi Market, Sarangpur	Surat	Gujarat	10 AM

20. Display detail of distributors which are active.

select *

from distributor

where status = 'true';



```
select *
from distributor
where status = 'true';
```

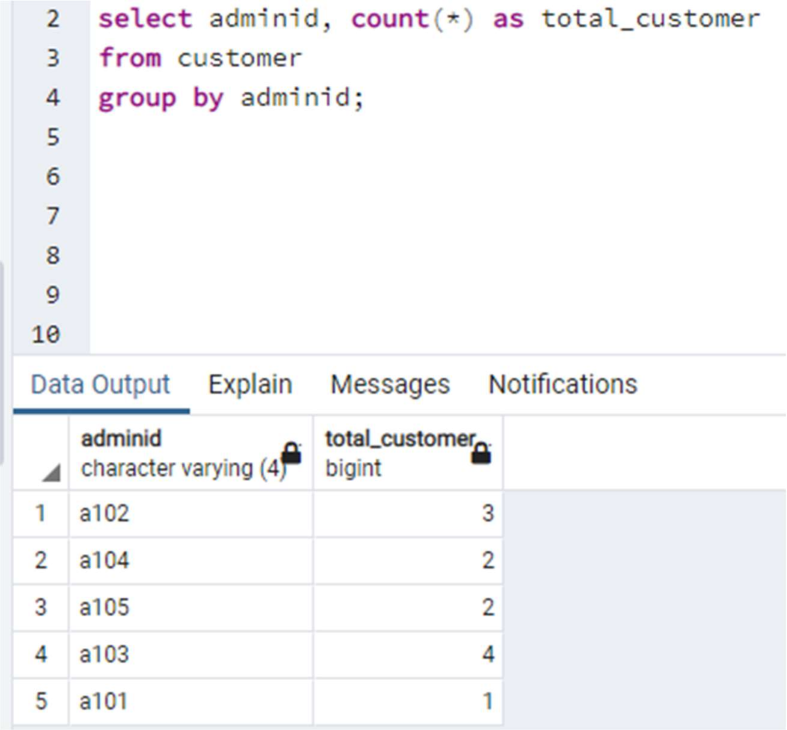
	distributorid [PK] character varying (4)	name character varying (20)	username character varying (30)	email character varying (254)	password character varying (30)	street character varying (50)	city character varying (50)
1	d102	Sunny Singh	me.ss	sunny@gmail.com	12345	A-87, mk park street	Surat
2	d104	Haresh Patel	haresh_patel	haresh@gmail.com	12345	T-123, sv school street	Surat
3	d101	Janak Raj	j_r	janak@gmail.com	12345	C-12, Link Street Road	Surat
4	d103	Pramod Mathur	mathur_pramod	pramod@gmail.com	12345	B-4, Dave Street	Surat

21. Count the total no. of customer handled by all the admin.

select adminid, count(*) as total_customer

from customer

group by adminid;



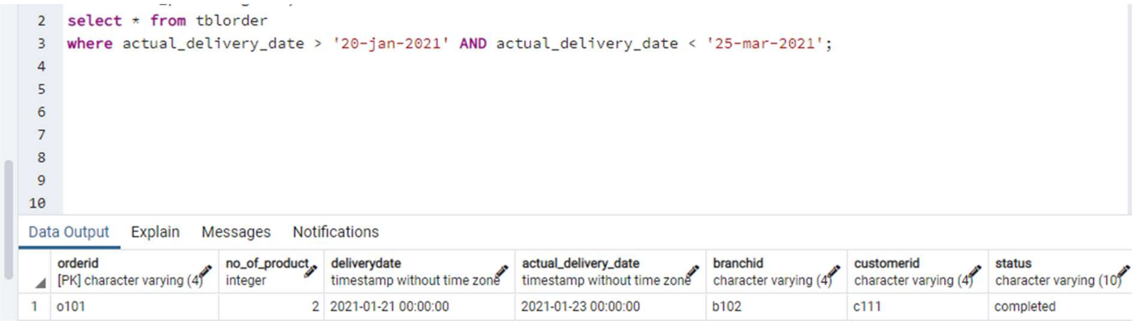
```
select adminid, count(*) as total_customer
from customer
group by adminid;
```

	adminid character varying (4)	total_customer bigint
1	a102	3
2	a104	2
3	a105	2
4	a103	4
5	a101	1

22. Display the details of order delivered at actual delivery date between '20-jan-2021' and '25-mar-2021'

select * from tblorder

where actual_delivery_date > '20-jan-2021' AND actual_delivery_date < '25-mar-2021';



```
select * from tblorder
where actual_delivery_date > '20-jan-2021' AND actual_delivery_date < '25-mar-2021';
```

	orderid [PK] character varying (4)	no_of_product integer	deliverydate timestamp without time zone	actual_delivery_date timestamp without time zone	branchid character varying (4)	customerid character varying (4)	status character varying (10)
1	o101	2	2021-01-21 00:00:00	2021-01-23 00:00:00	b102	c111	completed

23. Display the product detail on which there is discount more than 5 percentage.

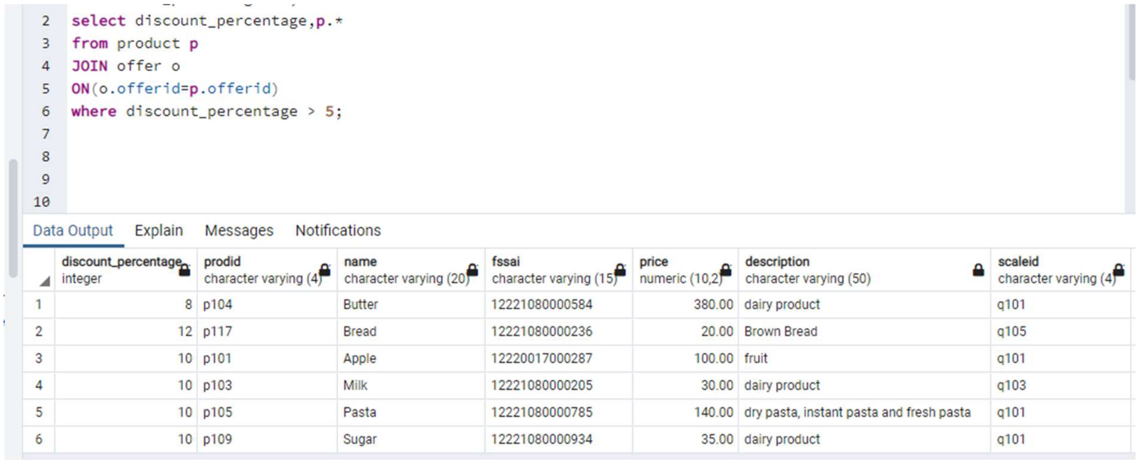
select p.*

from product p

JOIN offer o

ON(o.offerid=p.offerid)

where discount_percentage > 5;



```
select discount_percentage,p.*
from product p
JOIN offer o
ON(o.offerid=p.offerid)
where discount_percentage > 5;
```

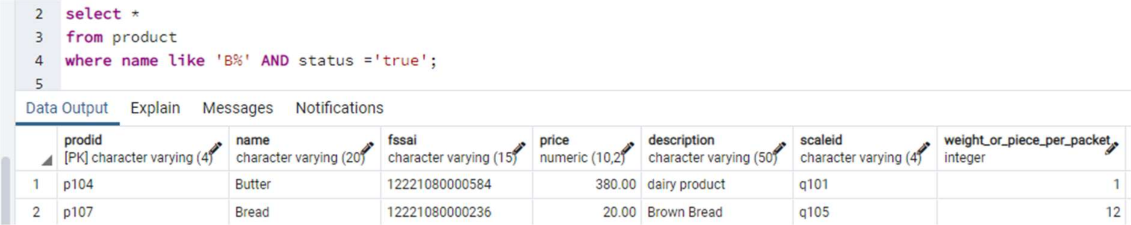
	discount_percentage integer	prodid character varying (4)	name character varying (20)	fssai character varying (15)	price numeric (10,2)	description character varying (50)	scaleid character varying (4)
1	8	p104	Butter	12221080000584	380.00	dairy product	q101
2	12	p117	Bread	12221080000236	20.00	Brown Bread	q105
3	10	p101	Apple	12220017000287	100.00	fruit	q101
4	10	p103	Milk	12221080000205	30.00	dairy product	q103
5	10	p105	Pasta	12221080000785	140.00	dry pasta, instant pasta and fresh pasta	q101
6	10	p109	Sugar	12221080000934	35.00	dairy product	q101

24. Display the detail of product name stating with 'B' and status ‘active’.

select *

from product

where name like 'B%' AND status ='true';



```
select *
from product
where name like 'B%' AND status ='true';
```

	prodid [PK] character varying (4)	name character varying (20)	fssai character varying (15)	price numeric (10,2)	description character varying (50)	scaleid character varying (4)	weight_or_piece_per_packet integer
1	p104	Butter	12221080000584	380.00	dairy product	q101	1
2	p107	Bread	12221080000236	20.00	Brown Bread	q105	12

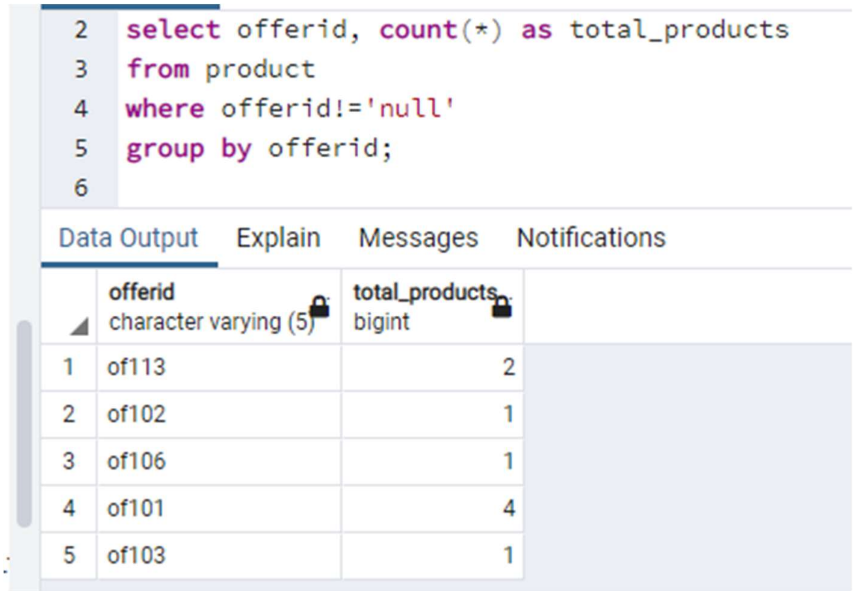
25. Count product for all the offers.

select offerid, count(*) as total_products

from product

where offerid!='null'

group by offerid;



```
select offerid, count(*) as total_products
from product
where offerid!='null'
group by offerid;
```

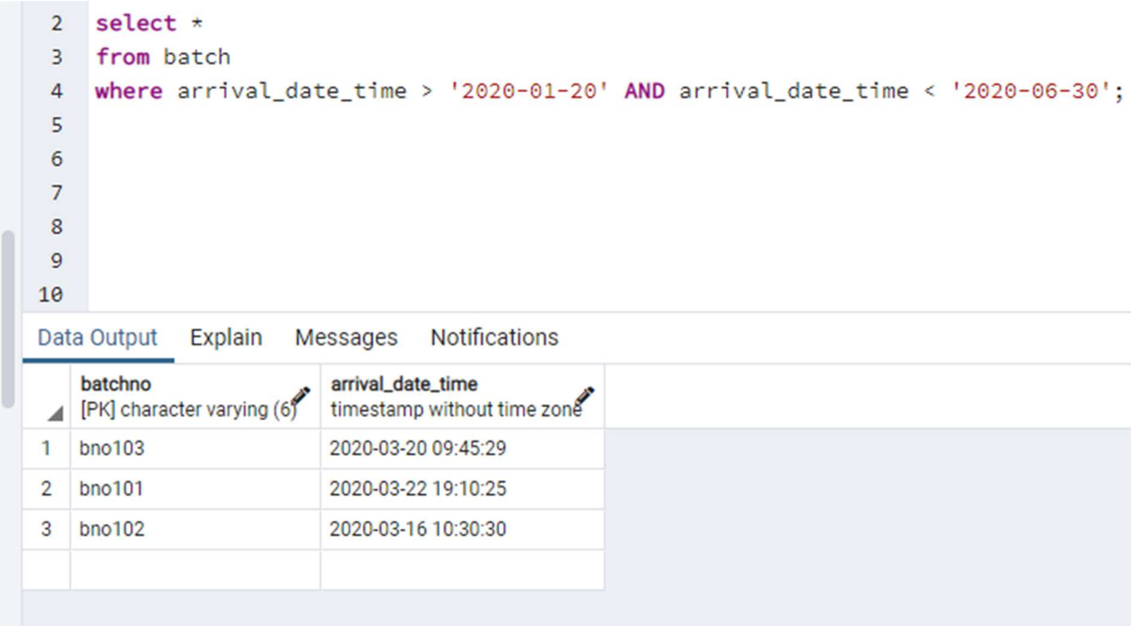
	offerid character varying (5)	total_products bigint
1	of113	2
2	of102	1
3	of106	1
4	of101	4
5	of103	1

26. Display the details of all the batch which were received between 2020-06-20 to 2020-06-30

select *

from batch

where arrival_date_time > '2020-01-20' AND arrival_date_time < '2020-06-30';



```
select *
from batch
where arrival_date_time > '2020-01-20' AND arrival_date_time < '2020-06-30';
```

	batchno [PK] character varying (6)	arrival_date_time timestamp without time zone
1	bno103	2020-03-20 09:45:29
2	bno101	2020-03-22 19:10:25
3	bno102	2020-03-16 10:30:30

SUMMARY

We have built an application that is user-friendly and can be accessed effortlessly. The firm's data includes distributors' data and information related to branches handled by them, and stock availability is collected. Customers can buy their needed products if available, and data related to their orders are stored. There are still some areas that have a future scope of development and can be implemented to improve the services provided by the system.