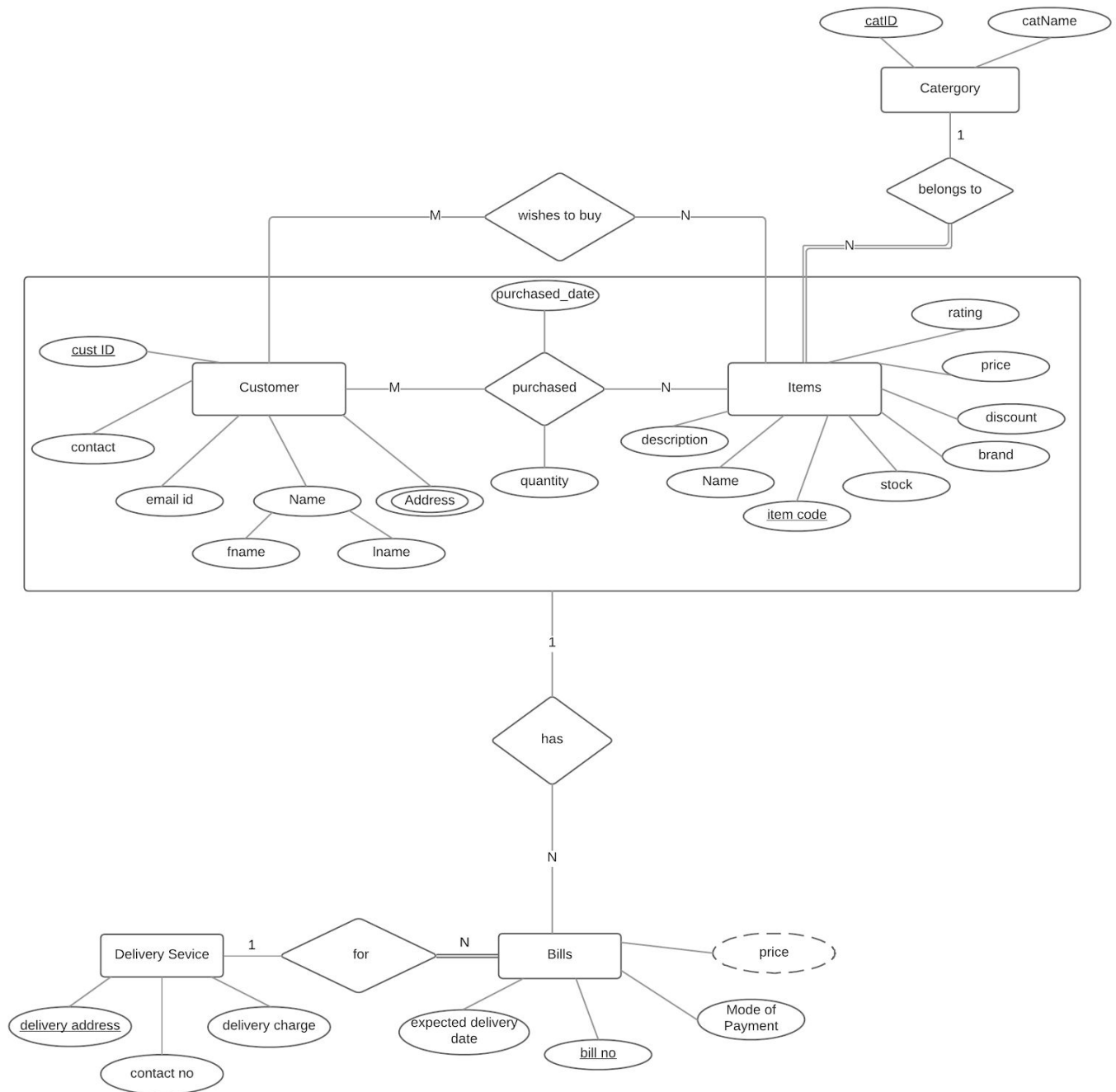
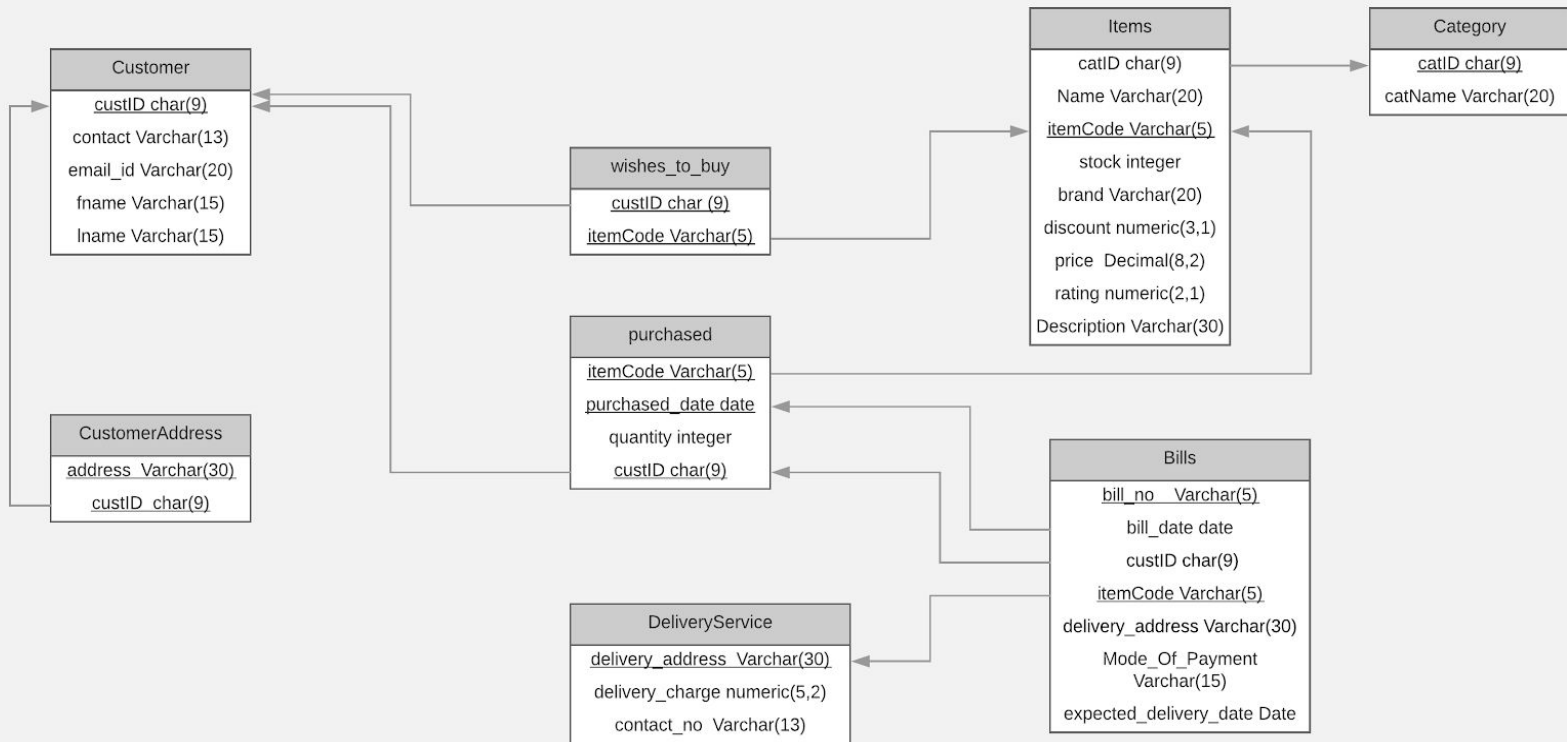


ER Diagram for online shopping database management (dealdone.com)



Relational Schema Diagram for dealdone.com



Queries for dealdone.com

List of relations			
Schema	Name	Type	Owner
dealdone	bills	table	201651019
dealdone	category	table	201651019
dealdone	customer	table	201651019
dealdone	customeraddress	table	201651019
dealdone	deliveryservice	table	201651019
dealdone	items	table	201651019
dealdone	purchased	table	201651019
dealdone	wishes_to_buy	table	201651019
(8 rows)			

bills

bill_no	custid	itemcode	delivery_address	bill_date	mode_of_payment	expected_delivery_date
10001	201801704	DS105	Kanpur	2018-01-01	Credit Card	2018-01-07
10002	201801701	BK101	Meerut	2018-01-05	Debit card	2018-01-08
10002	201801701	BK104	Meerut	2018-01-05	Debit card	2018-01-08
10003	201801701	TY102	Meerut	2018-02-02	COD	2018-02-06
10004	201801701	EA101	Meerut	2018-02-15	Credit Card	2018-02-18
10005	201801702	DS104	Kolkata	2018-02-20	Debit card	2018-02-23
10005	201801702	LU101	Kolkata	2018-02-20	Debit card	2018-02-23
10006	201801703	DS103	Varanasi	2018-03-09	COD	2018-03-11
10007	201801703	KA101	Varanasi	2018-03-16	Debit Card	2018-03-19
10007	201801703	KA102	Varanasi	2018-03-16	Debit Card	2018-03-19
10008	201801705	EA101	Hyderabad	2018-03-16	COD	2018-03-19
10009	201801706	KA103	Meerut	2018-04-01	Debit card	2018-04-03
10010	201801706	LU101	Meerut	2018-04-10	Credit Card	2018-04-12
10011	201801708	TY102	Varanasi	2018-04-11	Debit Card	2018-04-13
10011	201801708	DS101	Varanasi	2018-04-11	Debit Card	2018-04-13
10012	201801708	BK102	Varanasi	2018-04-28	Credit Card	2018-05-08
10013	201801710	MB101	Jabalpur	2018-04-20	Credit Card	2018-04-30
(17 rows)						

category

catid	catname
SP1111111	Sports
DS1111111	Dresses
BK1111111	Books
KA1111111	Kitchen Accessories
TY1111111	Toys
EA1111111	Electronics
LU1111111	Luggage
(7 rows)	

customer

custid	fname	lname	contact	email_id
201801701	Kirtika	Singhal	903265479	id1@gmail.com
201801702	Ishita	Das	903251478	id2@gmail.com
201801703	Mayank	Pathela	903253771	id3@gmail.com
201801704	Mahima	Arora	9032547103	md@gmail.com
201801705	Rachana	Soma	9433251897	rs@gmail.com
201801706	Pritha	Upadhyay	9433251890	pu@gmail.com
201801707	Vaaibhavi	Singh	9433251812	vs@gmail.com
201801708	Sandhya	Yadav	9433251813	sy@gmail.com
201801709	Monika	Phadnis	9433251814	mk@gmail.com
201801710	Parul	Soni	9433251815	psoni@gmail.com
201801711	Ajay	Guru	9433251816	ag@gmail.com
201801712	Mohak	Khare	9433251817	mk@gmail.com
201801713	Rohit	Sharma	9433251818	rsharma@gmail.com
201801714	Aashutosh	Rathi	9433251819	ar@gmail.com
201801715	Rahul	Singh	9433251820	rsingh@gmail.com
201801716	Anand	Menon	9433251821	am@gmail.com
201801717	Shruti	Rao	9433251822	srao@gmail.com
201801718	Pooja	Gurjar	9433251823	pooja1@gmail.com
201801719	Raju	Rastogi	9433251824	rr@gmail.com
201801720	Surabhi	Ramu	9433251825	ramu@gmail.com
201801721	Roshni	Ram	9433251826	rram@gmail.com
201801722	Sanjana	Dutta	9433251827	sdutta@gmail.com
201801723	Salman	Khan	9433251830	blackbuck@gmail.com
201801724	Shalini	Singhania	9433251828	shalini@gmail.com
201801725	Pratiti	Jain	9433251829	pjain@gmail.com

(25 rows)

wishes_to_buy

custid	itemcode
201801701	EA101
201801702	MB101
201801701	BK102
201801703	SP102
201801703	MB101
201801705	TY103
201801705	BK102

(7 rows)

customeraddress

custid	address
201801701	Meerut
201801702	Kolkata
201801702	Gandhinagar
201801703	Varanasi
201801704	Kanpur
201801704	Delhi
201801705	Hyderabad
201801706	Meerut
201801707	Lucknow
201801707	Gandhinagar
201801708	Varanasi
201801709	Mumbai
201801709	Gandhinagar
201801710	Jabalpur
201801711	Gwalior
201801711	Delhi
201801712	Jhansi
201801713	Nagpur
201801714	Nagpur
201801715	Jhansi
201801715	Ahmedabad
201801716	Meerut
201801717	Hyderabad
201801717	Bangalore
201801718	Jaipur
201801719	Ahmedabad
201801720	Kanpur
201801721	Kolkata
201801722	Mumbai
201801723	Lucknow
201801724	Gandhinagar
201801725	Kolkata

(32 rows)

deliveryservice

delivery_address	delivery_charge	contact_no
Mumbai	49.00	18001234
Kolkata	79.00	18002234
Chennai	49.00	18003234
Pune	49.00	18004234
Gandhinagar	89.00	18005234
Ahmedabad	59.00	18006234
Delhi	49.00	18007234
Jabalpur	79.00	18008234
Kanpur	59.00	18009234
Varanasi	79.00	18001134
Lucknow	79.00	18002134
Meerut	59.00	18002334
Hyderabad	29.00	18002434
Gwalior	49.00	18002534
Jhansi	39.00	18002634
Nagpur	29.00	18002734
Jaipur	49.00	18002834
Bangalore	19.00	18002934

(18 rows)

items

itemcode	name	catid	stock	brand	discount	price	rating	description
MB101	Iphone 7	EA1111111	8	apple	12.1	48000.00	4.6	5.5 inch screen gorilla glass
BK101	There Were None	BK1111111	10	Penguin	0.0	499.00	4.9	Murder Mystery
TY102	Mickey mouse	TY1111111	3	hamleys	30.2	1399.00	3.2	stuffed soft toy
SP101	Football	SP1111111	6	Nike	5.8	989.00	4.8	Limited Edition
LU101	Trolley Bag	LU1111111	11	Sky Bag	6.3	6071.00	4.6	23 inches Travel Bag
TY103	Minie mouse	TY1111111	2	hamleys	30.2	1299.00	3.4	stuffed soft toy
EA101	Laptop	EA1111111	3	DELL	10.0	75000.00	4.5	Black Colour i5 8GB
KA101	Knife	KA1111111	1	Prestige	5.0	236.00	4.9	Stainless Steel Blade
BK102	Alchemist	BK1111111	7	Rupa.Co	14.5	399.00	3.8	fantasy fiction
SP102	badminton racket	SP1111111	4	Reebok	4.8	1299.00	4.6	firm grip good quality
DS101	western gown	DS1111111	10	Madame	20.0	10000.00	4.0	party wear
DS102	black dress	DS1111111	5	Madame	20.0	599.00	4.0	party wear
DS103	deathnote tshirt	DS1111111	4	animania	10.0	699.00	4.2	round neck
DS104	natsu tshirt	DS1111111	3	animania	10.0	699.00	4.1	round neck
DS105	fairytale tshirt	DS1111111	4	animania	10.0	699.00	4.1	round neck
BK103	Kite Runner	BK1111111	3	Rupa.Co	24.5	499.00	4.8	fiction
BK104	The godfather	BK1111111	6	Penguin	27.0	499.00	4.7	fiction
KA102	Induction	KA1111111	4	Prestige	24.0	2999.00	4.5	long lasting
KA103	Electric Kettle	KA1111111	3	Phillips	30.8	1799.00	3.8	Stainless Steel

(19 rows)

purchased

itemcode	custid	quantity	purchased_date
EA101	201801705	1	2018-03-16
BK101	201801701	1	2018-01-05
TY102	201801701	3	2018-02-02
DS103	201801703	1	2018-03-09
KA101	201801703	2	2018-03-16
DS104	201801702	1	2018-02-20
LU101	201801702	1	2018-02-20
BK104	201801701	1	2018-01-05
KA103	201801706	2	2018-04-01
LU101	201801706	1	2018-04-10
EA101	201801701	1	2018-02-15
TY102	201801708	3	2018-04-11
DS101	201801708	2	2018-04-11
BK102	201801708	2	2018-04-28
KA102	201801703	1	2018-03-16
MB101	201801710	1	2018-04-20
DS105	201801704	2	2018-01-01
DS105	201801705	2	2018-01-01
DS105	201801707	2	2018-01-05
DS101	201801708	2	2018-01-21

(20 rows)

FUNCTIONAL DEPENDENCIES :

Customer(custID, contact, email_id, fname, lname){

 custID \rightarrow contact

 custID \rightarrow email_id

 custID \rightarrow fname

 custID \rightarrow lname

 Email_id \rightarrow custID

 Email_id \rightarrow fname

 Email_id \rightarrow lname

 Email_id \rightarrow contact

Key1: {custID}

Key2 : {email_id}

BCNF }

CustomerAddress(address, custID){

 Since, this relation has only two attributes, therefore it is in BCNF

Key : custID, address

BCNF }

wishes_to_buy(custID, itemcode){

Key: custID, itemcode

Since, this relation has only two attributes, therefore it is in BCNF }

Category(catID, catName){

 catID \rightarrow catName

Key : catID

BCNF }

Items(catID, Name, itemCode, stock, brand, discount, price, rating, Description){

 itemCode \rightarrow catID

 itemCode \rightarrow Name

 itemCode \rightarrow stock

 itemCode \rightarrow brand

 itemCode \rightarrow discount

 itemCode \rightarrow price

 itemCode \rightarrow rating

 itemCode \rightarrow Description

Key : itemCode

BCNF }

```

purchased( itemCode, purchased_date, quantity, custID){
    itemCode, purchased_date, custID → quantity
    Key : {itemCode, purchased_date, custID}
    BCNF
}

```

```

DeliveryService( delivery_address, delivery_charge, contact_no){
    Delivery_address → delivery_charge
    Delivery_address → contact_no
    Key : Delivery_address
    BCNF
}

```

```

Bills(bill_no, bill_date, custID, delivery_address, Mode_Of_Payment,
expected_delivery_date,itemCode) {
    Bill_no → custID
    Bill_no → delivery_address
    Bill_no → Mode_OF_Payment
    Bill_no → expected_delivery_date
    Bill_no → bill_date
    Key: {bill_no,itemCode}
    NOT IN BCNF. So, decomposed into bills1 and bills2 using BCNF decomposition
}

```

```

Bills1(bill_no, bill_date, custID, delivery_address, Mode_Of_Payment, expected_delivery_date){
    Bill_no → custID
    Bill_no → delivery_address
    Bill_no → Mode_OF_Payment
    Bill_no → expected_delivery_date
    Bill_no → bill_date
    Key: bill_no
    BCNF
}

```

```

Bills2(bill_no, itemCode){
    Since, this relation has only two attributes, therefore it is in BCNF.
    Key: bill_no, itemCode
}

```


SELECTION QUERIES :

1.Retrieve the total amount of items sold every month.

```
SELECT EXTRACT(month FROM purchased_date) AS month, round(sum(quantity*(price-(price*discount/100))),2) AS total_sale FROM purchased NATURAL JOIN items GROUP BY EXTRACT(month FROM purchased_date);
```

month	total_sale
1	20637.87
2	76747.13
3	70856.74
4	69982.14

(4 rows)

2. Find the Total Amount by per item sold every month.

```
SELECT EXTRACT(month FROM purchased_date) AS month, itemcode, round(sum(quantity*(price-price*discount/100)),2)AS sale FROM purchased NATURAL JOIN items GROUP BY (EXTRACT(month FROM purchased_date), itemcode) ORDER BY EXTRACT(month FROM purchased_date);
```

month	itemcode	sale
1	DS105	3774.60
1	BK101	499.00
1	DS101	16000.00
1	BK104	364.27
2	EA101	67500.00
2	TY102	2929.51
2	DS104	629.10
2	LU101	5688.53
3	EA101	67500.00
3	DS103	629.10
3	KA101	448.40
3	KA102	2279.24
4	TY102	2929.51
4	BK102	682.29
4	MB101	42192.00
4	LU101	5688.53
4	KA103	2489.82
4	DS101	16000.00

(18 rows)

3. Retrieve number of customers in a particular city.

```
SELECT DISTINCT count(custid), address FROM customeraddress GROUP BY address;
```

count	address
2	Delhi
3	Meerut
2	Jhansi
1	Jabalpur
1	Jaipur
2	Varanasi
1	Bangalore
2	Ahmedabad
4	Gandhinagar
3	Kolkata
2	Mumbai
2	Nagpur
2	Kanpur
2	Hyderabad
2	Lucknow
1	Gwalior

(16 rows)

4. Retrieve category having maximum sale in terms of quantity.

```
SELECT DISTINCT ON(date_part)* FROM(SELECT DISTINCT  
catid,catname,sum(quantity), EXTRACT(month FROM purchased_date)  
FROM category NATURAL JOIN items NATURAL JOIN purchased GROUP BY  
catid, EXTRACT(month FROM purchased_date) ORDER BY  
date_part,sum desc)AS f WHERE date_part=4;
```

catid	catname	sum	date_part
TY1111111	Toys	3	4

(1 row)

5. Retrieve most valuable customer per month in terms of amount

```
SELECT DISTINCT ON(month) * FROM (SELECT custid,  
sum(quantity),round(sum(quantity*(price-price*discount/100)),2),  
EXTRACT(month FROM purchased_date) AS month FROM purchased  
NATURAL JOIN items GROUP BY custid, EXTRACT(month FROM  
purchased_date) ORDER BY month,round desc) AS f;
```

custid	sum	round	month
201801708	2	16000.00	1
201801701	4	70429.51	2
201801705	1	67500.00	3
201801710	1	42192.00	4

(4 rows)

6. Retrieve item details for range 500-1000 of price.

```
SELECT * FROM items WHERE price between 500 and 1000;
```

itemcode	name	catid	stock	brand	discount	price	rating	description
SP101	Football	SP1111111	6	Nike	5.8	989.00	4.8	Limited Edition
DS102	black dress	DS1111111	5	Madame	20.0	599.00	4.0	party wear
DS103	deathnote tshirt	DS1111111	4	animania	10.0	699.00	4.2	round neck
DS104	natsu tshirt	DS1111111	3	animania	10.0	699.00	4.1	round neck
DS105	fairytale tshirt	DS1111111	4	animania	10.0	699.00	4.1	round neck

(5 rows)

7. List of items in wishlist of maximum customers.

```
SELECT itemcode, name, count(itemcode) AS qty
FROM wishes_to_buy NATURAL JOIN items GROUP BY itemcode, name
HAVING count(itemcode) IN (SELECT max(a) FROM (SELECT
count(itemcode) AS a FROM wishes_to_buy NATURAL JOIN items GROUP
BY name,itemcode order by a) AS b);
```

itemcode	name	qty
BK102	Alchemist	2
MB101	Iphone 7	2
(2 rows)		

8. Total amount on a bill including delivery charge.

```
SELECT bill_no,custid,(A+delivery_charge) AS total_charge FROM
deliveryservice NATURAL JOIN (select bill_no, custid, delivery_address,
round(sum(quantity*(price-price*discount/100)),2) AS A FROM bills
NATURAL JOIN items NATURAL JOIN purchased GROUP BY bill_no, custid,
delivery_address) AS B ORDER BY bill_no;
```

bill_no	custid	total_charge
10001	201801704	1317.20
10002	201801701	922.27
10003	201801701	2988.51
10004	201801701	67559.00
10005	201801702	6396.63
10006	201801703	708.10
10007	201801703	2806.64
10008	201801705	67529.00
10009	201801706	2548.82
10010	201801706	5747.53
10011	201801708	35008.51
10012	201801708	761.29
10013	201801710	42271.00
10014	201801705	1287.20
10015	201801707	1337.20
10016	201801708	32079.00
(16 rows)		

9. List Details of items HAVING stock > 6.

```
SELECT * FROM items WHERE stock>6;
```

```
201651029=> SELECT * FROM items WHERE stock>6;
itemcode | name | catid | stock | brand | discount | price | rating | description
-----+-----+-----+-----+-----+-----+-----+-----+-----
MB101 | Iphone 7 | EA1111111 | 8 | apple | 12.1 | 48000.00 | 4.6 | 5.5 inch screen gorilla glass
BK101 | There Were None | BK1111111 | 10 | Penguin | 0.0 | 499.00 | 4.9 | Murder Mystery
LU101 | Trolley Bag | LU1111111 | 11 | Sky Bag | 6.3 | 6071.00 | 4.6 | 23 inches Travel Bag
BK102 | Alchemist | BK1111111 | 7 | Rupa.Co | 14.5 | 399.00 | 3.8 | fantasy fiction
DS101 | western gown | DS1111111 | 10 | Madame | 20.0 | 10000.00 | 4.0 | party wear
(5 rows)
```

10. Retrieve customer_id who has purchased from maximum number of categories.

```
SELECT custid,count(DISTINCT catid) FROM purchased NATURAL JOIN
items GROUP BY custid HAVING count(DISTINCT catid) IN(SELECT
max(c.count) FROM (SELECT custid,count(DISTINCT catid) FROM
purchased NATURAL JOIN items GROUP BY custid order by count)AS c);
```

```
custid | count
-----+-----
201801701 | 3
201801708 | 3
(2 rows)
```

11. Bills corresponding to a customer id='201801701' including delivery charge.

```
SELECT bill_no,custid,(A+delivery_charge) AS total_charge FROM
deliveryservice NATURAL JOIN (select bill_no, custid, delivery_address,
round(sum(quantity*(price-price*discount/100)),2) AS A FROM bills
NATURAL JOIN items NATURAL JOIN purchased GROUP BY bill_no,
custid, delivery_address having custid = '201801701') AS B;
```

bill_no	custid	total_charge
10002	201801701	922.27
10003	201801701	2988.51
10004	201801701	67559.00
(3 rows)		

12. Number of bills for a customer

```
SELECT count(DISTINCT bill_no) AS number_of_bills, custid,
round(sum(quantity*(price-price*discount/100)),2) AS total_value
FROM bills NATURAL JOIN items NATURAL JOIN purchased
GROUP BY custid ORDER BY number_of_bills desc;
```

number_of_bills	custid	total_value
3	201801708	67611.80
3	201801701	71292.78
2	201801705	68758.20
2	201801703	3356.74
2	201801706	8178.34
1	201801710	42192.00
1	201801707	1258.20
1	201801702	6317.63
1	201801704	1258.20
(9 rows)		

13. Most frequent customer

```
SELECT count(DISTINCT bill_no) AS number_of_bills, custid,  
round(sum(quantity*(price-price*discount/100)),2) AS total_value  
FROM bills NATURAL JOIN items NATURAL JOIN purchased  
GROUP BY custid HAVING(count(DISTINCT bill_no)) IN  
(SELECT max(number_of_bills) FROM (SELECT count(DISTINCT bill_no)  
AS  
number_of_bills,custid,round(sum(quantity*(price-price*discount/100)),2)  
AS total_value FROM bills NATURAL JOIN items NATURAL JOIN  
purchased GROUP BY custid ORDER BY number_of_bills desc) AS f);
```

number_of_bills	custid	total_value
3	201801701	71292.78
3	201801708	67611.80

(2 rows)

14. List of all items of bill_no = '10005'

```
SELECT bill_no,itemcode,name,round((price-(price*discount/100)),2)  
AS price,custid FROM bills NATURAL JOIN items WHERE bill_no='10005'  
GROUP BY bill_no,itemcode,name,custid,price,discount order by bill_no;
```

bill_no	itemcode	name	price	custid
10005	DS104	natsu tshirt	629.10	201801702
10005	LU101	Trolley Bag	5688.53	201801702

(2 rows)

15. Retrieve address of the customer having customer id='201801702'

```
SELECT custid,address FROM customeraddress  
WHERE custid='201801702';
```

custid	address
201801702	Kolkata
201801702	Gandhinagar

(2 rows)

16. List all items order by price,category.

```
SELECT * FROM items ORDER BY price,catid;
```

itemcode	name	catid	stock	brand	discount	price	rating	description
KA101	Knife	KA1111111	1	Prestige	5.0	236.00	4.9	Stainless Steel Blade
BK102	Alchemist	BK1111111	7	Rupa.Co	14.5	399.00	3.8	fantasy fiction
BK103	Kite Runner	BK1111111	3	Rupa.Co	24.5	499.00	4.8	fiction
BK101	There Were None	BK1111111	10	Penguin	0.0	499.00	4.9	Murder Mystery
BK104	The godfather	BK1111111	6	Penguin	27.0	499.00	4.7	fiction
DS102	black dress	DS1111111	5	Madame	20.0	599.00	4.0	party wear
DS103	deathnote tshirt	DS1111111	4	animania	10.0	699.00	4.2	round neck
DS105	fairytale tshirt	DS1111111	4	animania	10.0	699.00	4.1	round neck
DS104	natsu tshirt	DS1111111	3	animania	10.0	699.00	4.1	round neck
SP101	Football	SP1111111	6	Nike	5.8	989.00	4.8	Limited Edition
SP102	badminton racket	SP1111111	4	Reebok	4.8	1299.00	4.6	firm grip good quality
TY103	Minie mouse	TY1111111	2	hamleys	30.2	1299.00	3.4	stuffed soft toy
TY102	Mickey mouse	TY1111111	3	hamleys	30.2	1399.00	3.2	stuffed soft toy
KA103	Electric Kettle	KA1111111	3	Phillips	30.8	1799.00	3.8	Stainless Steel
KA102	Induction	KA1111111	4	Prestige	24.0	2999.00	4.5	long lasting
LU101	Trolley Bag	LU1111111	11	Sky Bag	6.3	6071.00	4.6	23 inches Travel Bag
DS101	western gown	DS1111111	10	Madame	20.0	10000.00	4.0	party wear
MB101	Iphone 7	EA1111111	8	apple	12.1	48000.00	4.6	5.5 inch screen gorilla glass
EA101	Laptop	EA1111111	3	DELL	10.0	75000.00	4.5	Black Colour i5 8GB

(19 rows)

17. Retrieve most selling item month-wise.

```
SELECT DISTINCT ON(month)* FROM (SELECT itemcode, name,
EXTRACT(month FROM purchased_date) AS month, sum(quantity) AS
total FROM purchased NATURAL JOIN items GROUP BY EXTRACT(month
FROM purchased_date),itemCode,name ORDER BY month, total desc) AS
f;
```

itemcode	name	month	total
DS105	fairytale tshirt	1	6
TY102	Mickey mouse	2	3
KA101	Knife	3	2
TY102	Mickey mouse	4	3

(4 rows)

18. Maximum orders from a given address

```
SELECT count(DISTINCT bill_no) AS number_of_orders, delivery_address
FROM bills GROUP BY delivery_address HAVING count(DISTINCT bill_no)
IN(SELECT max(number_of_orders) FROM (SELECT count(DISTINCT
bill_no) AS number_of_orders FROM bills GROUP BY delivery_address
ORDER BY number_of_orders desc) AS f);
```

number_of_orders	delivery_address
5	Meerut
5	Varanasi

(2 rows)

19. Retrieve catid from which most of the item are purchased WHERE custid='201801701'.

```
SELECT custid, catid, count(catid) FROM purchased NATURAL JOIN items
WHERE custid='201801701' GROUP BY catid, custid HAVING count(catid)
IN(SELECT max(b.count) FROM (SELECT custid, count(catid), catid
FROM purchased NATURAL JOIN items WHERE custid= '201801701'
GROUP BY custid, catid ORDER BY count) AS b);
```

custid	catid	count
201801701	BK1111111	2

(1 row)

20. List custid and custname who haven't purchased any items.

```
SELECT custid,fname,lname,contact FROM customer  
EXCEPT (SELECT custid,fname,lname,contact FROM customer NATURAL  
JOIN purchased);
```

custid	fname	lname	contact
201801718	Pooja	Gurjar	9433251823
201801709	Monika	Phadnis	9433251814
201801714	Aashutosh	Rathi	9433251819
201801723	Salman	Khan	9433251830
201801711	Ajay	Guru	9433251816
201801716	Anand	Menon	9433251821
201801722	Sanjana	Dutta	9433251827
201801720	Surabhi	Ramu	9433251825
201801715	Rahul	Singh	9433251820
201801721	Roshni	Ram	9433251826
201801717	Shruti	Rao	9433251822
201801724	Shalini	Singhania	9433251828
201801712	Mohak	Khare	9433251817
201801719	Raju	Rastogi	9433251824
201801725	Pratiti	Jain	9433251829
201801713	Rohit	Sharma	9433251818

(16 rows)

21. Daily Sales

```
SELECT bill_date,round(SUM(quantity*(price-price*discount/100)),2)  
AS total_sale FROM bills NATURAL JOIN purchased NATURAL JOIN items  
GROUP BY(bill_date) ORDER BY total_sale desc;
```

bill_date	total_sale
2018-03-16	70227.64
2018-02-15	67500.00
2018-04-20	42192.00
2018-04-11	34929.51
2018-01-21	32000.00
2018-02-20	6317.63
2018-04-10	5688.53
2018-02-02	2929.51
2018-01-01	2516.40
2018-04-01	2489.82
2018-01-05	2121.47
2018-04-28	682.29
2018-03-09	629.10

(13 rows)