E-COMMERCE

By:

16BCE0237	Aayush Kapur
16BCE0640	Nirut Gupta
16BCE0785	Lakshay Arora

Prof. Sharmila Banu K.

Slot: D2

Database Management Systems

CSE2004

VIT®
UNIVERSITY
(Estd. u/s 3 of UGC Act 1956)

ACKNOWLEDGEMENTS

We are very grateful to Prof. Sharmila Banu K, our project guide, without whose guidance we wouldn’t have completed this project. We would also like to thank VIT University management for providing us this opportunity to undertake this course and provide us with the facilities to finish this project.

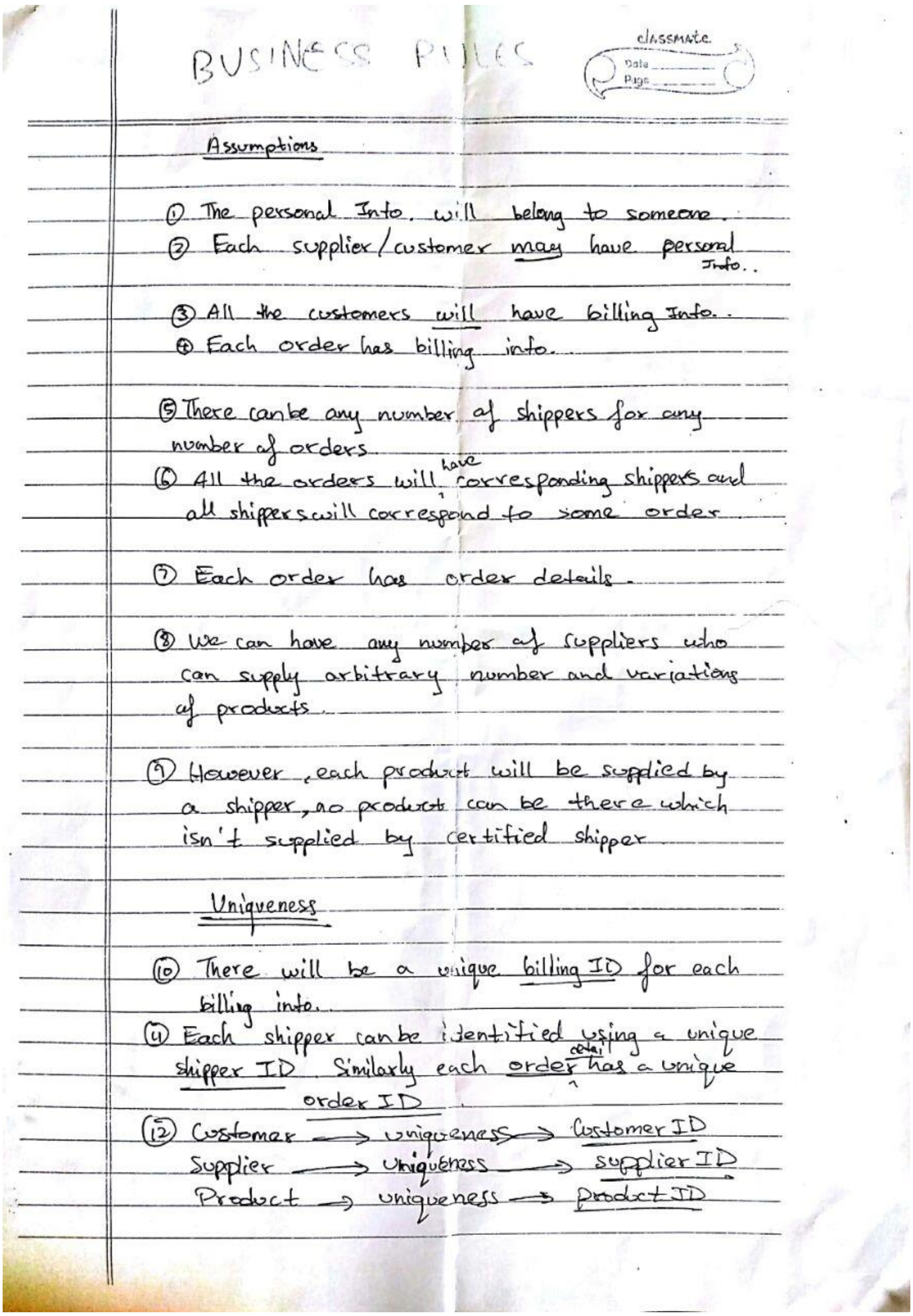
Sign:

16BCE0237 AAYUSH KAPUR

16BCE0640 NIRUT GUPTA

16BCE0785 LAKSHAY ARORA

BUSINESS RULES



BUSINESS RULES

classmate
Date
Page

Assumptions

① The personal Info. will belong to someone.
② Each supplier/customer may have personal Info..

③ All the customers will have billing Info..
④ Each order has billing info.

⑤ There can be any number of shippers for any number of orders.
⑥ All the orders will have corresponding shippers and all shippers will correspond to some order.

⑦ Each order has order details.

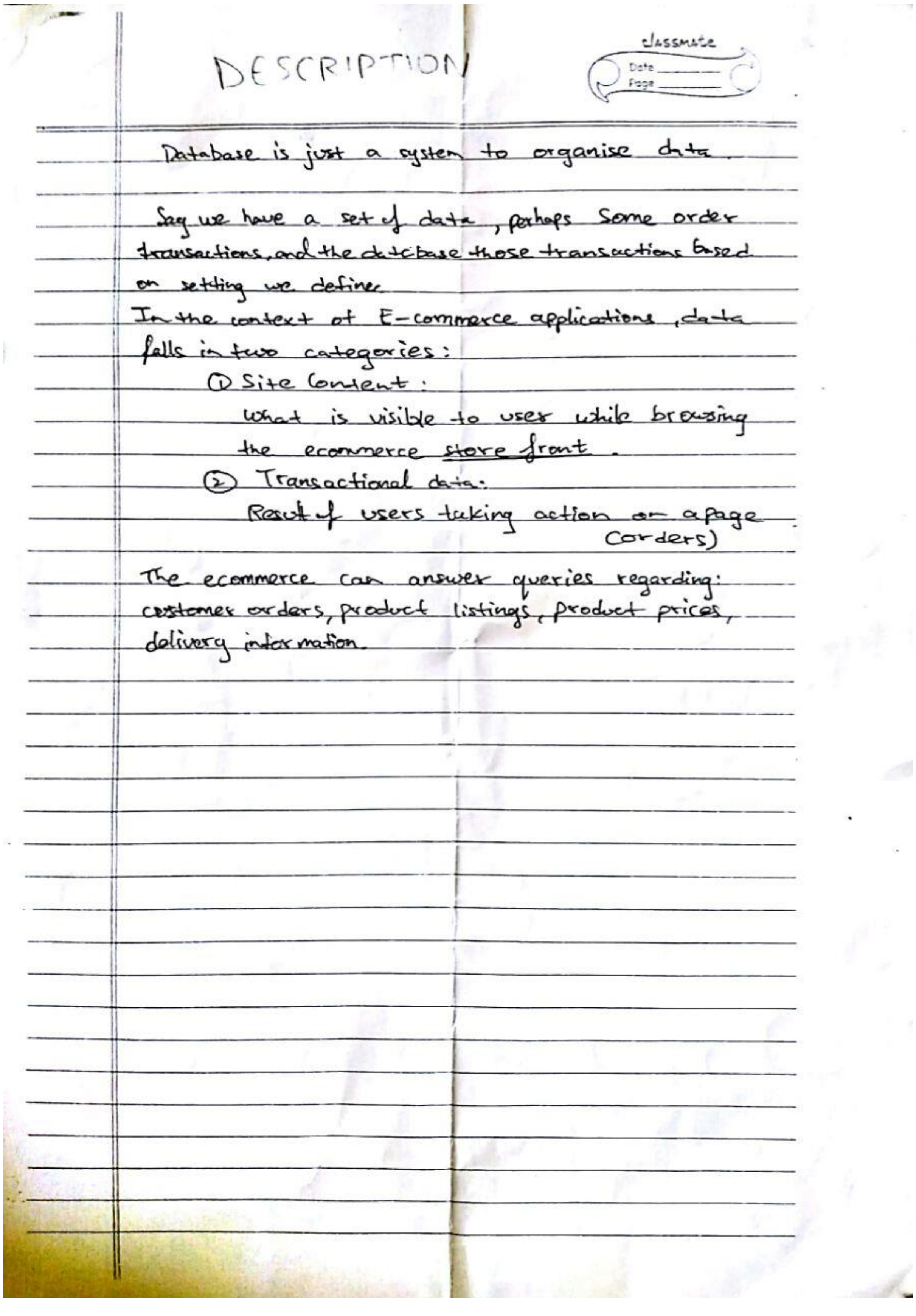
⑧ We can have any number of suppliers who can supply arbitrary number and variations of products.

⑨ However, each product will be supplied by a shipper, no product can be there which isn't supplied by certified shipper.

Uniqueness

⑩ There will be a unique billing ID for each billing info.
⑪ Each shipper can be identified using a unique shipper ID. Similarly each order detail has a unique order ID.
⑫ Customer → uniqueness → Customer ID
Supplier → Uniqueness → supplier ID
Product → uniqueness → product ID

DESCRIPTION



classmate

Date

Page

DESCRIPTION

Database is just a system to organise data.

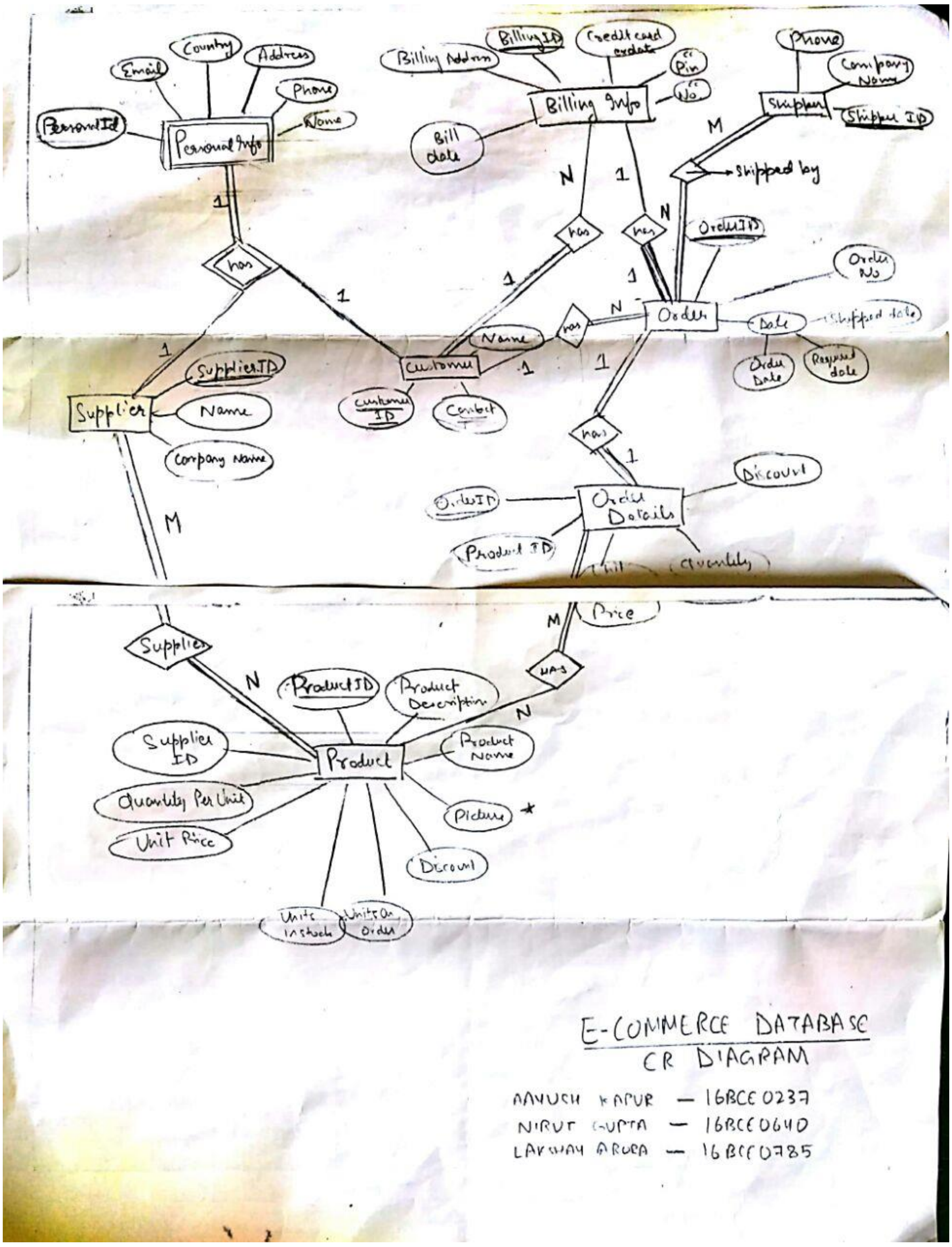
Say we have a set of data, perhaps Some order transactions, and the database those transactions based on setting we define.

In the context of E-commerce applications, data falls in two categories:

1. Site Content:
   What is visible to user while browsing the ecommerce <u>store front</u>.
2. Transactional data:
   Result of users taking action on a page (orders)

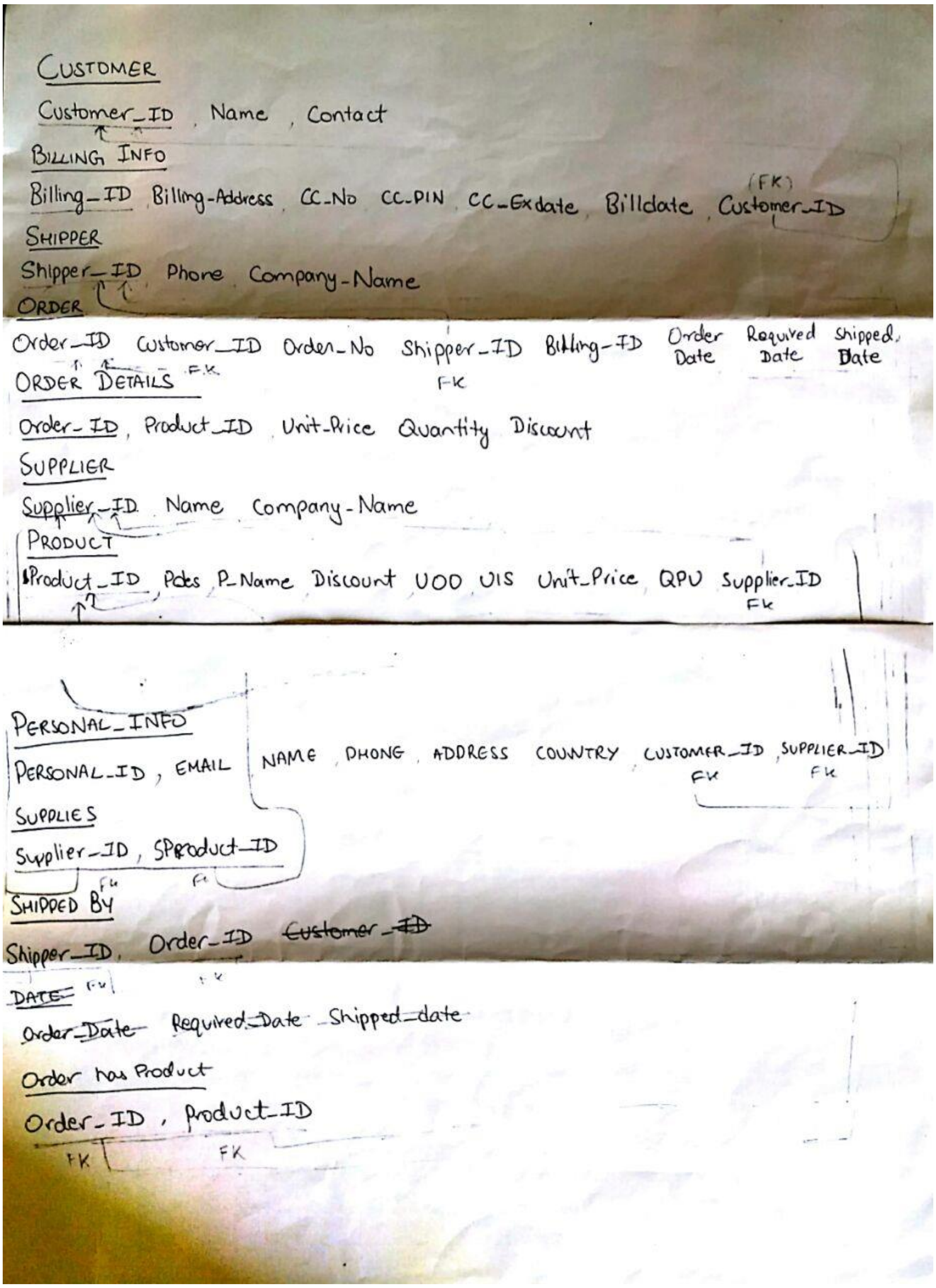
The ecommerce can answer queries regarding: customer orders, product listings, product prices, delivery information.

ER- DIAGRAM



Country
Email
Address
Phone
PersonalId
Name
Personal Info
1
has
1
Billing Address
BillingID
Credit card expdate
Pin
No.
Billing Info
Bill date
N
1
has
has
N
Phone
Company Name
Shipper
Shipper ID
M
Shipped by
OrderID
Order No
1
1
N
has
Order
Date
Shipped date
Order Date
Required date
Name
Customer
Customer ID
Contact
1
1
Supplier ID
Supplier
Name
Company Name
has
1
Discount
OrderID
Order Details
Product ID
Quantity
M
Price
M
Supplier
HAS
N
N
ProductID
Product Description
Supplier ID
Product
Product Name
Quantity Per Unit
Unit Price
Picture
Discount
Units In stock
Units On Order
E-COMMERCE DATABASE
ER DIAGRAM
AAYUSH KAPUR — 16BCE0237
NIRUT GUPTA — 16BCE0640
LAKSHAY ARORA — 16BCE0785

TABLE SCHEMA



CUSTOMER

Customer_ID , Name , Contact

BILLING INFO

Billing_ID , Billing-Address , CC_No , CC_PIN , CC_Exdate , Billdate , Customer_ID (FK)

SHIPPER

Shipper_ID , Phone , Company-Name

ORDER

Order_ID , Customer_ID , Order_No , Shipper_ID , Billing_ID , Order Date , Required Date , Shipped Date

FK FK

ORDER DETAILS

Order_ID , Product_ID , Unit_Price , Quantity , Discount

SUPPLIER

Supplier_ID , Name , Company-Name

PRODUCT

Product_ID , Pcks , P_Name , Discount , UOO , UIS , Unit_Price , QPU , Supplier_ID
FK

PERSONAL_INFO

PERSONAL_ID , EMAIL , NAME , PHONE , ADDRESS , COUNTRY , CUSTOMER_ID , SUPPLIER_ID
FK FK

SUPPLIES

Supplier_ID , SProduct_ID
FK FK

SHIPPED BY

Shipper_ID , Order_ID ~~Customer_ID~~
FK FK

~~DATE~~

Order_Date Required_Date Shipped_date

Order has Product

Order_ID , Product_ID
FK FK

Using MySQL to build up the database

CREATING A TABLE

```
create table order_has_product (order_id varchar(40), product_id varchar(40));

create table shipped_by (shipper_id varchar(40), order_id varchar(40));

create table supplies (supplier_id varchar(40),product_id varchar(40));

create table personal_info (personal_id varchar(40),email varchar(40), first_name
varchar(40),contact_no int(20),address varchar(100),country varchar(20),customer_id
varchar(40),supplier_id varchar(40));

create table customer(customer_id varchar(20), name varchar(50), contact int(20))
CHARACTER SET utf8 COLLATE utf8_general_ci;

create table orders
(order_id varchar(20),
customer_id varchar(20),
order_no int(5),
shipper_id varchar(20),
billing_id varchar(20),
order_date date,
req_date date,
shipped_date date);

create table order_details(
order_id varchar(20),
product_id varchar(20),
unit_price int(5),
quantity int(5),
order_discount float);

create table billing_info(billing_id varchar(20), billing_address varchar(50), cc_no int(20),
cc_pin int(4), cc_exdate date, billdate date,customer_id varchar(20));

create table shipper(shipper_id varchar(20), phone int(20), company_name varchar(50));

create table supplier
(supplier_id varchar(20),
name varchar(20),
company_name varchar(20));

create table product(product_id varchar(20), product_Des varchar(100), product_n
varchar(100), discount float, units_ordered int(5),units_stock int(20),unit_price int(20),
quan_per_unit int(20), supplier_id varchar(20));
```

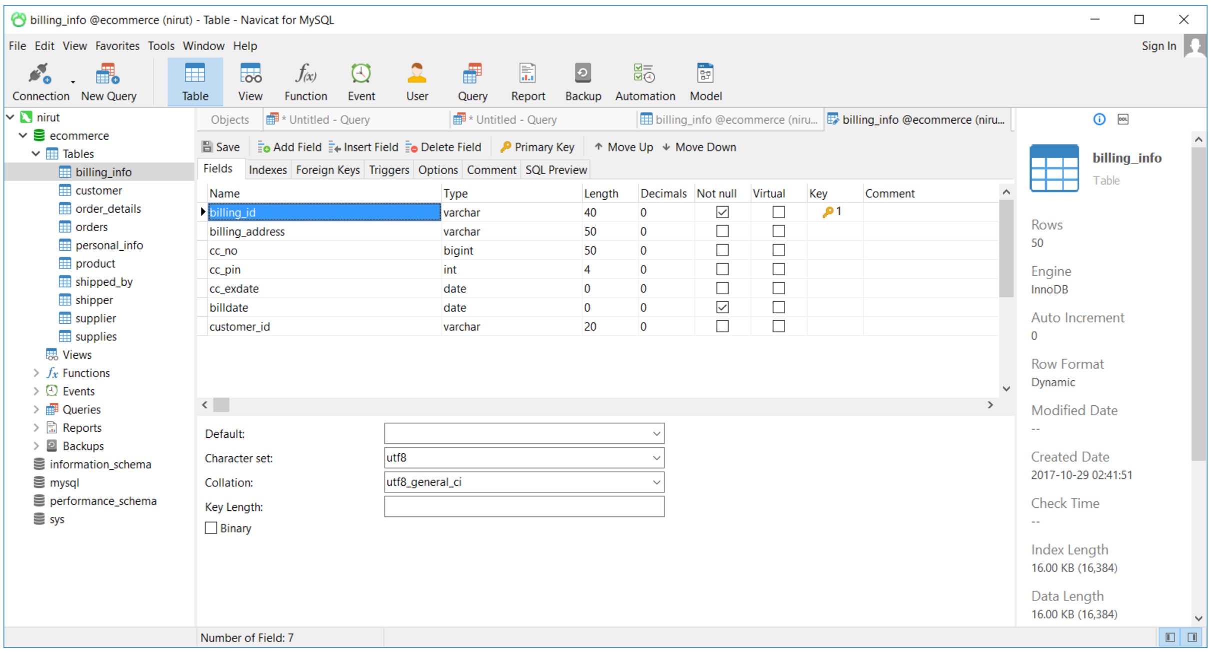
ADDING CONTRAINTS

```
alter table billing_info modify column cc_no bigint(50);
alter table billing_info modify column billing_id bigint(50);
alter table billing_info modify billing_id varchar(40);
alter table orders add constraint p_orders primary key(order_id);
alter table order_details add constraint p_order_details primary key(order_id,product_id);
alter table supplier add constraint p_supplier primary key(supplier_id);
alter table orders modify order_no int(5) not null;
alter table order_details modify unit_price int(5) not null;
alter table order_details add constraint check(unit_price>=0);
alter table order_details add constraint check(quantity>=0);
alter table order_details add constraint check(order_discount>=0);
alter table orders add constraint check(discount>=0);
alter table order_details alter column order_discount set default 0;
alter table shipped_by add constraint sho_p primary key(shipper_id,order_id);
alter table supplies add constraint ss_p primary key(supplier_id,product_id);
alter table product add constraint pro_p primary key(product_id);
alter table product modify  units_ordered int(5) not null;
alter table product modify  unit_price int(20) not null;
alter table product modify  product_id varchar(20) not null;
alter table personal_info add constraint per_p primary key(personal_id);
alter table product alter column discount set default 0;
alter table order_details alter column discount set default 0;
alter table product add constraint check(quan_per_unit>=0);
alter table product add constraint check(unit_price>=0);
alter table product add constraint check(units_stock>=0);
alter table customer add constraint p_customer primary key(customer_id);
alter table billing_info add constraint p_billing_info primary key(billing_id);
alter table shipper add constraint p_shipper primary key(shipper_id);
alter table billing_info modify billdate date not null;
alter table shipper modify phone int(20) not null;
alter table shipper modify company_name varchar(50) not null;
```

ADDING FOREIGN KEYS

```
alter table billing_info add constraint pk_billing_info foreign key(customer_id) references
customer(customer_id)on delete cascade;
alter table supplies add constraint fk_supplier_id foreign key(supplier_id) references
supplier(supplier_id) on delete cascade;
alter table shipped_by add constraint fr_2 foreign key(order_id) references orders(order_id)
on delete cascade;
alter table product add constraint foreign key (supplier_id) references supplier(supplier_id)
on delete set null ;
alter table orders add constraint foreign key (shipper_id) references shipper(shipper_id) on
delete set null;
alter table personal_info add constraint foreign key (customer_id) references
customer(customer_id) on delete cascade;
alter table supplies add constraint foreign key(product_id) references product(product_id) on
delete cascade;
alter table orders add constraint foreign key(customer_id) references customer(customer_id)
on delete cascade;
alter table personal_info add constraint foreign key(supplier_id) references
supplier(supplier_id) on delete cascade;
alter table shipped_by add constraint fr_1 foreign key (shipper_id) references
shipper(shipper_id) on delete cascade;
alter table order_details add constraint foreign key(order_id) references orders(order_id) on
delete cascade;
alter table
order_details add constraint foreign key(product_id) references
product(product_id) on delete cascade;
alter table
orders add constraint foreign key(billing_id) references
billing_info(billing_id) on delete cascade;
```

TABLE DETAILS OF BILLING_INFO


billing_info @ecommerce (nirut) - Table - Navicat for MySQL

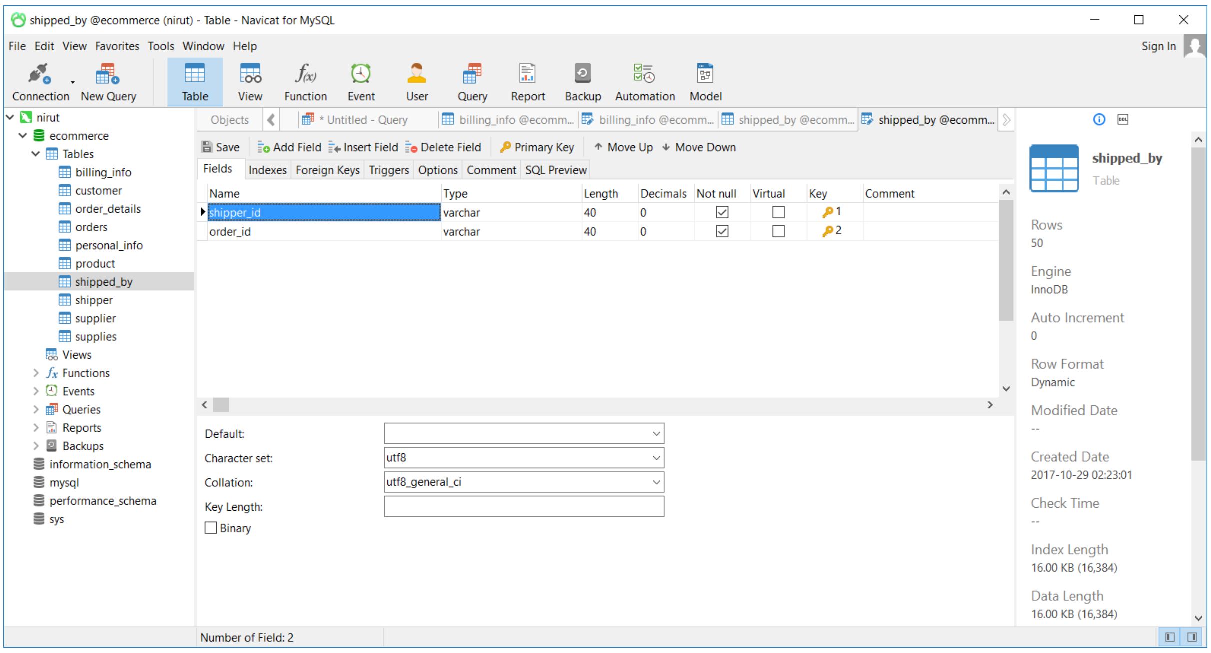
Name	Type	Length	Decimals	Not null	Virtual	Key	Comment
billing_id	varchar	40	0	☑	☐	🔑 1	
billing_address	varchar	50	0	☐	☐		
cc_no	bigint	50	0	☐	☐		
cc_pin	int	4	0	☐	☐		
cc_exdate	date	0	0	☐	☐		
billdate	date	0	0	☑	☐		
customer_id	varchar	20	0	☐	☐		

Default:
Character set: utf8
Collation: utf8_general_ci
Key Length:
☐ Binary

Number of Field: 7

billing_info
Table
Rows
50
Engine
InnoDB
Auto Increment
0
Row Format
Dynamic
Modified Date
--
Created Date
2017-10-29 02:41:51
Check Time
--
Index Length
16.00 KB (16,384)
Data Length
16.00 KB (16,384)

TABLE DETAILS OF SHIPPED_BY


shipped_by @ecommerce (nirut) - Table - Navicat for MySQL

Name	Type	Length	Decimals	Not null	Virtual	Key	Comment
shipper_id	varchar	40	0	☑	☐	🔑 1	
order_id	varchar	40	0	☑	☐	🔑 2	

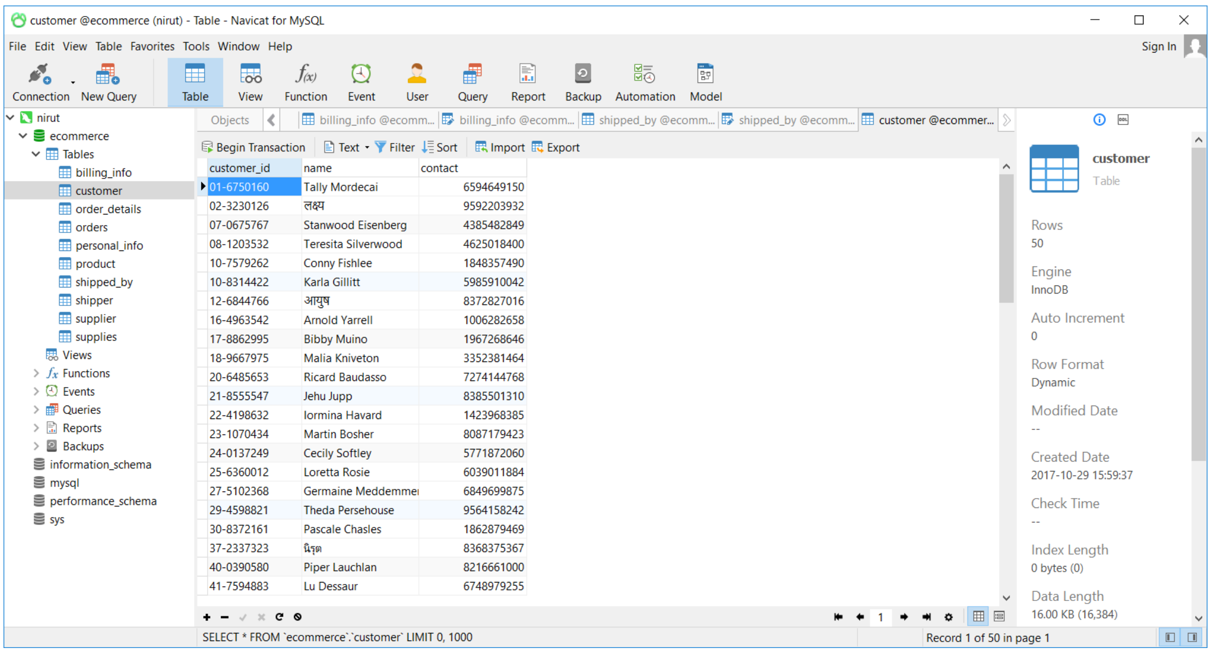
Default:
Character set: utf8
Collation: utf8_general_ci
Key Length:
☐ Binary

Number of Field: 2

shipped_by
Table
Rows
50
Engine
InnoDB
Auto Increment
0
Row Format
Dynamic
Modified Date
--
Created Date
2017-10-29 02:23:01
Check Time
--
Index Length
16.00 KB (16,384)
Data Length
16.00 KB (16,384)

MULTILINGUAL INSERTION

```
update customer
set name=N'นิรุต'
where name like 'Issy%';

update customer
set name=N'आयुष'
where name like 'Euell%';

update customer
set name=N'लक्ष्य'
where name like 'Madeleine%';

update customer
set name=N'李重伟'
where name like 'Sax%';

update customer
set name=N'владимир'
where name like 'Roze%';

update customer
set name=N'مرحبا'
where name like 'Rikki%';
```



customer_id	name	contact
01-6750160	Tally Mordecai	6594649150
02-3230126	लक्ष्य	9592203932
07-0675767	Stanwood Eisenberg	4385482849
08-1203532	Teresita Silverwood	4625018400
10-7579262	Conny Fishlee	1848357490
10-8314422	Karla Gillitt	5985910042
12-6844766	आयुष	8372827016
16-4963542	Arnold Yarrell	1006282658
17-8862995	Bibby Muino	1967268646
18-9667975	Malia Kniveton	3352381464
20-6485653	Ricard Baudasso	7274144768
21-8555547	Jehu Jupp	8385501310
22-4198632	Iormina Havard	1423968385
23-1070434	Martin Bosher	8087179423
24-0137249	Cecily Softley	5771872060
25-6360012	Loretta Rosie	6039011884
27-5102368	Germaine Meddemme	6849699875
29-4598821	Theda Persehouse	9564158242
30-8372161	Pascale Chasles	1862879469
37-2337323	นิรุต	8368375367
40-0390580	Piper Lauchlan	8216661000
41-7594883	Lu Dessaur	6748979255

customer
Table

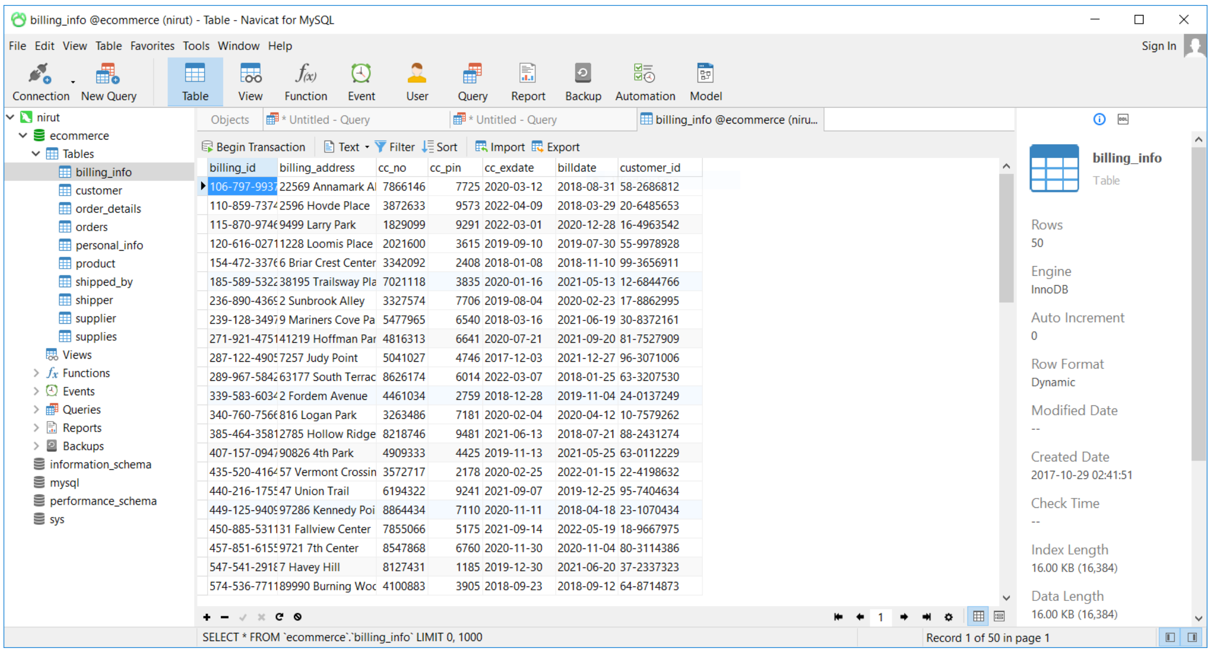
Rows	50
Engine	InnoDB
Auto Increment	0
Row Format	Dynamic
Modified Date	--
Created Date	2017-10-29 15:59:37
Check Time	--
Index Length	0 bytes (0)
Data Length	16.00 KB (16,384)

```
SELECT * FROM `ecommerce`.`customer` LIMIT 0, 1000
```

Record 1 of 50 in page 1

BULK INSERT USING CSV

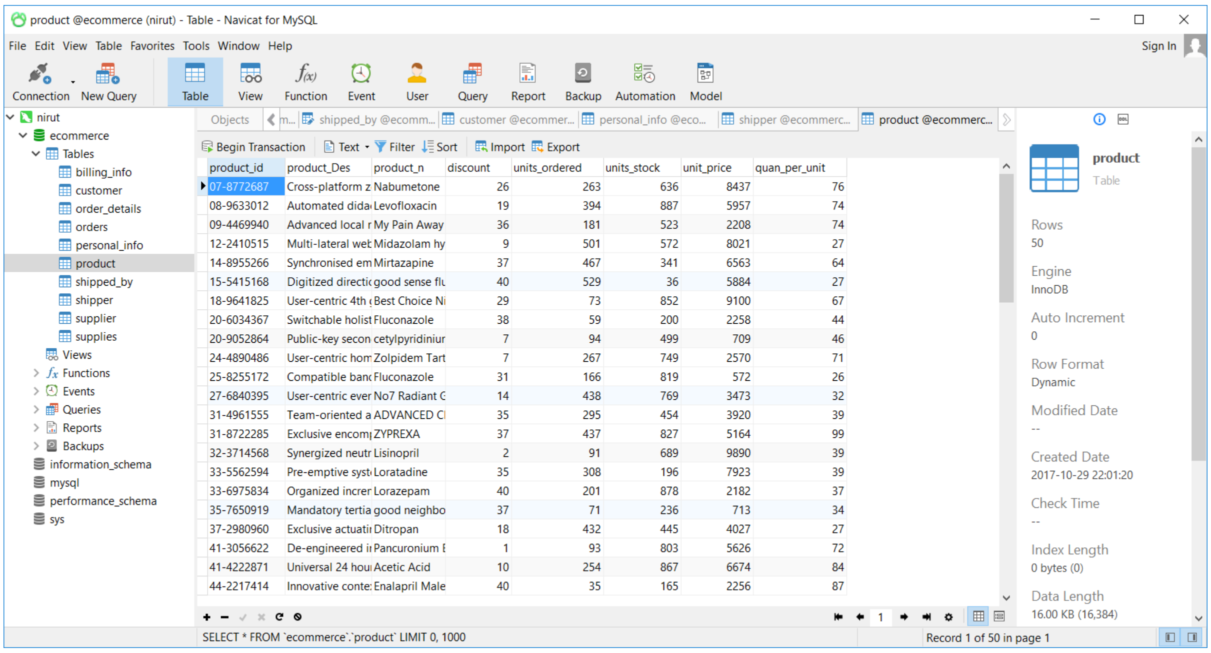
LOAD DATA LOCAL INFILE 'file address' INTO TABLE customer
FIELDS TERMINATED BY ',' ENCLOSED BY '"'
LINES TERMINATED BY ‘\r\n'
IGNORE 1 LINES;


billing_info @ecommerce (nirut) - Table - Navicat for MySQL

billing_id	billing_address	cc_no	cc_pin	cc_exdate	billdate	customer_id
106-797-993	22569 Annamark Al	7866146	7725	2020-03-12	2018-08-31	58-2686812
110-859-737	2596 Hovde Place	3872633	9573	2022-04-09	2018-03-29	20-6485653
115-870-974	9499 Larry Park	1829099	9291	2022-03-01	2020-12-28	16-4963542
120-616-0271	1228 Loomis Place	2021600	3615	2019-09-10	2019-07-30	55-9978928
154-472-337	6 Briar Crest Center	3342092	2408	2018-01-08	2018-11-10	99-3656911
185-589-532	38195 Trailsway Pla	7021118	3835	2020-01-16	2021-05-13	12-6844766
236-890-436	2 Sunbrook Alley	3327574	7706	2019-08-04	2020-02-23	17-8862995
239-128-349	9 Mariners Cove Pa	5477965	6540	2018-03-16	2021-06-19	30-8372161
271-921-4751	41219 Hoffman Par	4816313	6641	2020-07-21	2021-09-20	81-7527909
287-122-490	7257 Judy Point	5041027	4746	2017-12-03	2021-12-27	96-3071006
289-967-584	63177 South Terrac	8626174	6014	2022-03-07	2018-01-25	63-3207530
339-583-603	2 Fordem Avenue	4461034	2759	2018-12-28	2019-11-04	24-0137249
340-760-756	816 Logan Park	3263486	7181	2020-02-04	2020-04-12	10-7579262
385-464-3581	2785 Hollow Ridge	8218746	9481	2021-06-13	2018-07-21	88-2431274
407-157-094	90826 4th Park	4909333	4425	2019-11-13	2021-05-25	63-0112229
435-520-416	57 Vermont Crossin	3572717	2178	2020-02-25	2022-01-15	22-4198632
440-216-175	47 Union Trail	6194322	9241	2021-09-07	2019-12-25	95-7404634
449-125-940	97286 Kennedy Poi	8864434	7110	2020-11-11	2018-04-18	23-1070434
450-885-5311	31 Fallview Center	7855066	5175	2021-09-14	2022-05-19	18-9667975
457-851-615	9721 7th Center	8547868	6760	2020-11-30	2020-11-04	80-3114386
547-541-291	7 Havey Hill	8127431	1185	2019-12-30	2021-06-20	37-2337323
574-536-7711	89990 Burning Woc	4100883	3905	2018-09-23	2018-09-12	64-8714873

billing_info
Table
Rows
50
Engine
InnoDB
Auto Increment
0
Row Format
Dynamic
Modified Date
--
Created Date
2017-10-29 02:41:51
Check Time
--
Index Length
16.00 KB (16,384)
Data Length
16.00 KB (16,384)

SELECT * FROM `ecommerce`.`billing_info` LIMIT 0, 1000
Record 1 of 50 in page 1


product @ecommerce (nirut) - Table - Navicat for MySQL

product_id	product_Des	product_n	discount	units_ordered	units_stock	unit_price	quan_per_unit
07-8772687	Cross-platform z	Nabumetone	26	263	636	8437	76
08-9633012	Automated dida	Levofloxacin	19	394	887	5957	74
09-4469940	Advanced local r	My Pain Away	36	181	523	2208	74
12-2410515	Multi-lateral wet	Midazolam hy	9	501	572	8021	27
14-8955266	Synchronised em	Mirtazapine	37	467	341	6563	64
15-5415168	Digitized directi	good sense flu	40	529	36	5884	27
18-9641825	User-centric 4th	Best Choice Ni	29	73	852	9100	67
20-6034367	Switchable holist	Fluconazole	38	59	200	2258	44
20-9052864	Public-key secon	cetylpyridiniur	7	94	499	709	46
24-4890486	User-centric hom	Zolpidem Tart	7	267	749	2570	71
25-8255172	Compatible banc	Fluconazole	31	166	819	572	26
27-6840395	User-centric ever	No7 Radiant G	14	438	769	3473	32
31-4961555	Team-oriented a	ADVANCED CI	35	295	454	3920	39
31-8722285	Exclusive encom	ZYPREXA	37	437	827	5164	99
32-3714568	Synergized neutr	Lisinopril	2	91	689	9890	39
33-5562594	Pre-emptive syst	Loratadine	35	308	196	7923	39
33-6975834	Organized increr	Lorazepam	40	201	878	2182	37
35-7650919	Mandatory tertia	good neighbo	37	71	236	713	34
37-2980960	Exclusive actuati	Ditropan	18	432	445	4027	27
41-3056622	De-engineered ir	Pancuronium E	1	93	803	5626	72
41-4222871	Universal 24 hou	Acetic Acid	10	254	867	6674	84
44-2217414	Innovative conte:	Enalapril Male	40	35	165	2256	87

product
Table
Rows
50
Engine
InnoDB
Auto Increment
0
Row Format
Dynamic
Modified Date
--
Created Date
2017-10-29 22:01:20
Check Time
--
Index Length
0 bytes (0)
Data Length
16.00 KB (16,384)

SELECT * FROM `ecommerce`.`product` LIMIT 0, 1000
Record 1 of 50 in page 1

QUERY FOR DATA COMPRESSION

Alter table customer row_format=compressed;

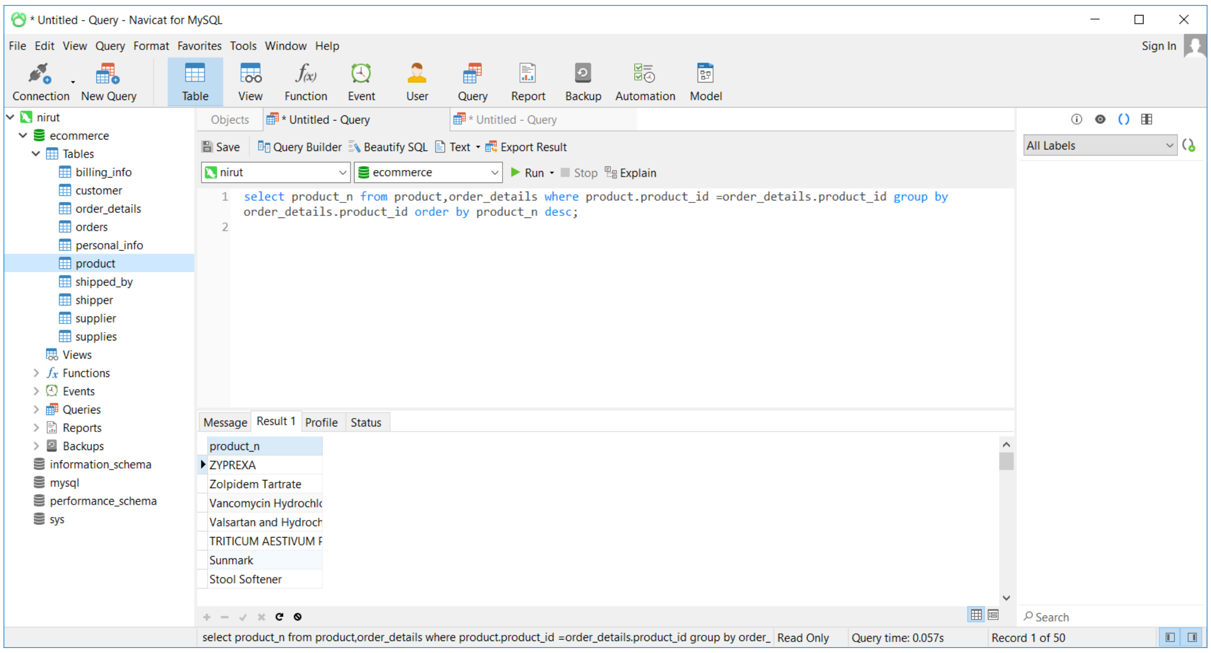
QUERY FOR SECURITY

GEANT SELECT ON *.* TO ‘username’ @ ‘localhost’ IDENTIFIED BY ‘password’;

QUERIES GIVEN IN REVIEW 2

select product_n from product,order_details where product.product_id=order_details.product_id

group by order_details.product_id order by product_n desc;



```
select product_n from product,order_details where product.product_id =order_details.product_id group by
order_details.product_id order by product_n desc;
```

```
product_n
ZYPREXA
Zolpidem Tartrate
Vancomycin Hydrochlc
Valsartan and Hydrocl
TRITICUM AESTIVUM F
Sunmark
Stool Softener
```

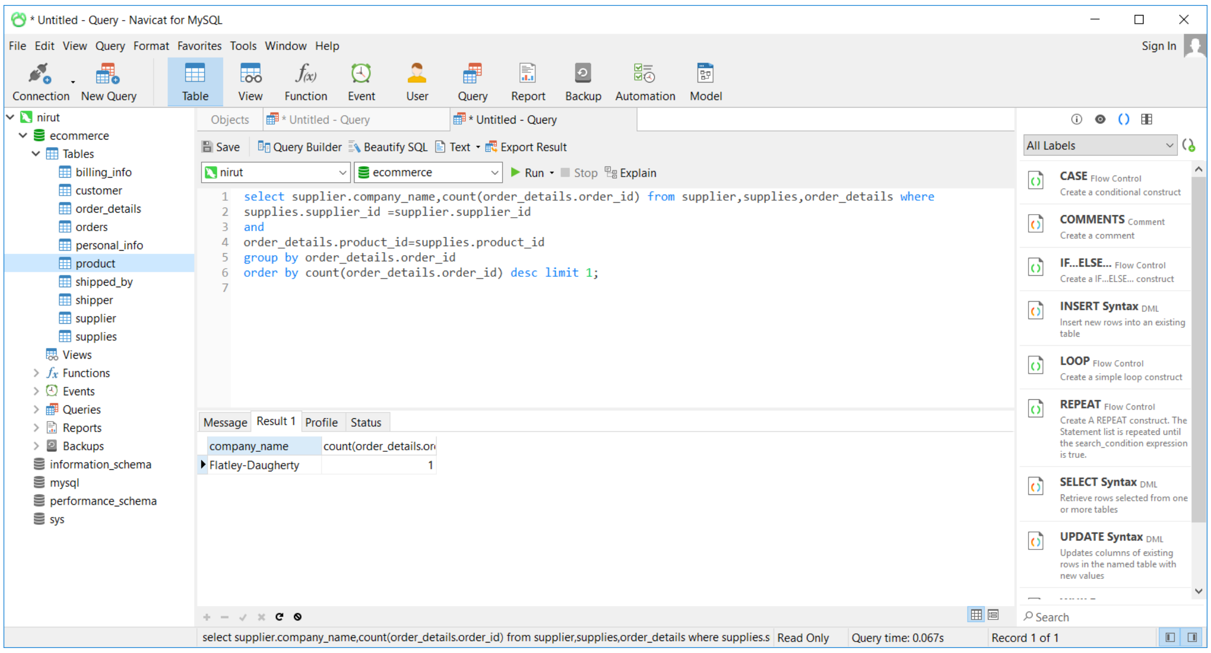
```
select product_n from product,order_details where product.product_id =order_details.product_id group by order_   Read Only   Query time: 0.057s   Record 1 of 50
```

select supplier.company_name,count(order_details.order_id) from supplier,supplies,order_details where

supplies.supplier_id =supplier.supplier_id and

order_details.product_id=supplies.product_id

group by order_details.order_id order by count(order_details.order_id) desc limit 1;

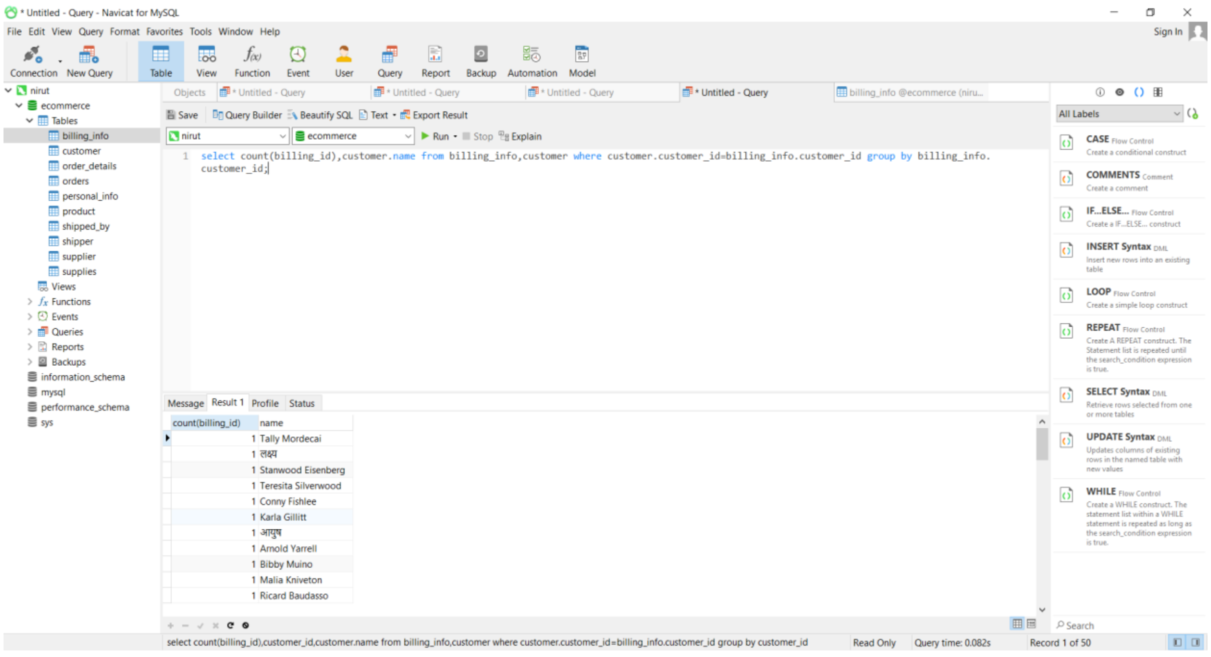


```
select supplier.company_name,count(order_details.order_id) from supplier,supplies,order_details where
supplies.supplier_id =supplier.supplier_id
and
order_details.product_id=supplies.product_id
group by order_details.order_id
order by count(order_details.order_id) desc limit 1;
```

```
company_name        count(order_details.or
Flatley-Daugherty                        1
```

```
select supplier.company_name,count(order_details.order_id) from supplier,supplies,order_details where supplies.s   Read Only   Query time: 0.067s   Record 1 of 1
```

select count(billing_id),customer.name from billing_info,customer where customer.customer_id=billing_info.customer_id group by billing_info.customer_id;



```
select count(billing_id),customer.name from billing_info,customer where customer.customer_id=billing_info.customer_id group by billing_info.
customer_id;
```

count(billing_id)	name
1	Tally Mordecai
1	लक्ष्य
1	Stanwood Eisenberg
1	Teresita Silverwood
1	Conny Fishlee
1	Karla Gillitt
1	आयुष
1	Arnold Yarrell
1	Bibby Muino
1	Malia Kniveton
1	Ricard Baudasso

```
select count(billing_id),customer_id,customer.name from billing_info,customer where customer.customer_id=billing_info.customer_id group by customer_id    Read Only    Query time: 0.082s    Record 1 of 50
```

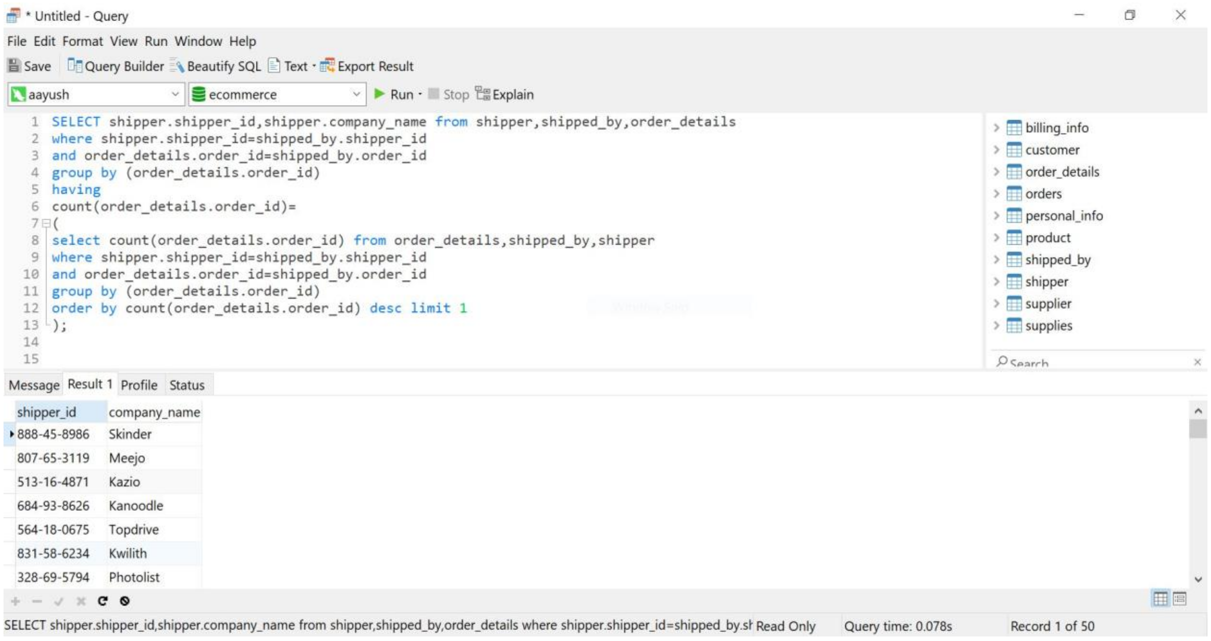
select shipper.shipper_id,shipper.company_name from shipper,shipped_by,order_details where shipper.shipper_id=shipped_by.shipper_id

and order_details.order_id=shipped_by.order_id

group by (order_details.order_id)having count(order_details.order_id)=(select count(order_details.order_id) from order_details,shipped_by,shipper where shipper.shipper_id=shipped_by.shipper_id and order_details.order_id=shipped_by.order_id

group by (order_details.order_id)

order by count(order_details.order_id) desc limit 1);



```
SELECT shipper.shipper_id,shipper.company_name from shipper,shipped_by,order_details
where shipper.shipper_id=shipped_by.shipper_id
and order_details.order_id=shipped_by.order_id
group by (order_details.order_id)
having
count(order_details.order_id)=
(
select count(order_details.order_id) from order_details,shipped_by,shipper
where shipper.shipper_id=shipped_by.shipper_id
and order_details.order_id=shipped_by.order_id
group by (order_details.order_id)
order by count(order_details.order_id) desc limit 1
);
```

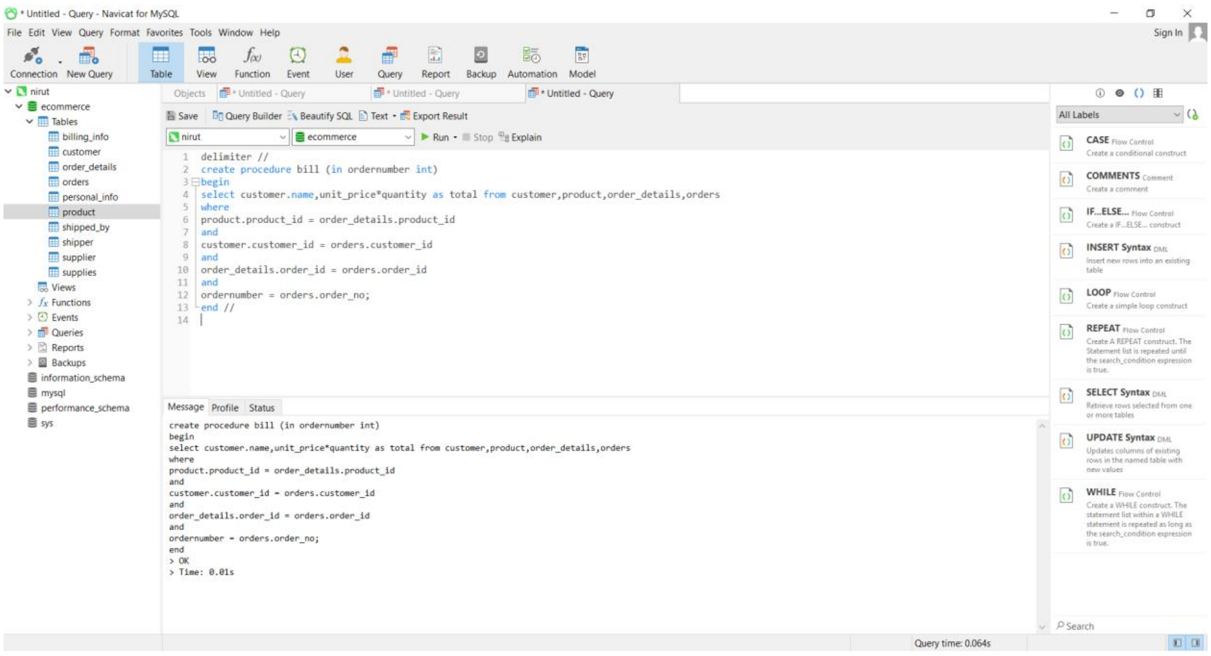
shipper_id	company_name
888-45-8986	Skinder
807-65-3119	Meejo
513-16-4871	Kazio
684-93-8626	Kanoodle
564-18-0675	Topdrive
831-58-6234	Kwilith
328-69-5794	Photolist

```
SELECT shipper.shipper_id,shipper.company_name from shipper,shipped_by,order_details where shipper.shipper_id=shipped_by.sh    Read Only    Query time: 0.078s    Record 1 of 50
```

PL/SQL QUERIES

QUERY 1:

delimiter //
create procedure bill (in ordernumber int)
begin
select customer.name,unit_price*quantity as total from customer,product,order_details,orders
where
product.product_id = order_details.product_id
and
customer.customer_id = orders.customer_id
and
order_details.order_id = orders.order_id
and
ordernumber = orders.order_no;
end //



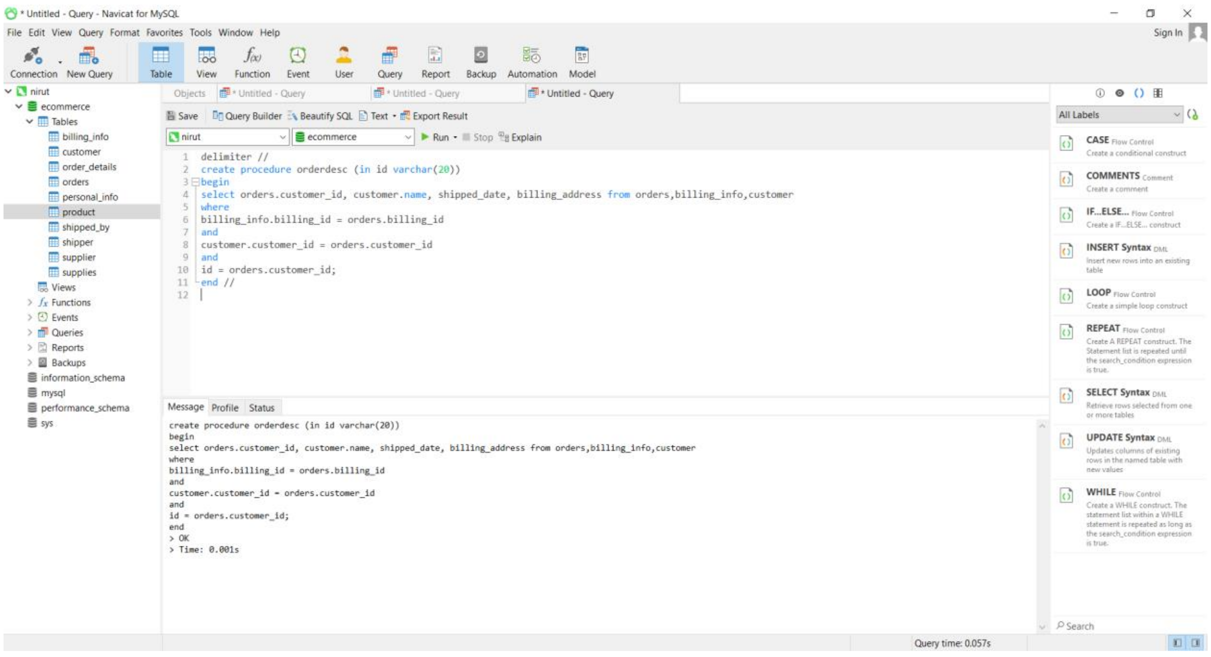
```
delimiter //
create procedure bill (in ordernumber int)
begin
select customer.name,unit_price*quantity as total from customer,product,order_details,orders
where
product.product_id = order_details.product_id
and
customer.customer_id = orders.customer_id
and
order_details.order_id = orders.order_id
and
ordernumber = orders.order_no;
end //
```

```
create procedure bill (in ordernumber int)
begin
select customer.name,unit_price*quantity as total from customer,product,order_details,orders
where
product.product_id = order_details.product_id
and
customer.customer_id = orders.customer_id
and
order_details.order_id = orders.order_id
and
ordernumber = orders.order_no;
end
> OK
> Time: 0.01s
```

Query time: 0.064s

QUERY 2:

```
delimiter //
create procedure orderdesc (in id varchar(20))
begin
select orders.customer_id, customer.name, shipped_date, billing_address from
orders,billing_info,customer
where
billing_info.billing_id = orders.billing_id
and
customer.customer_id = orders.customer_id
and
id = orders.customer_id;
end //
```



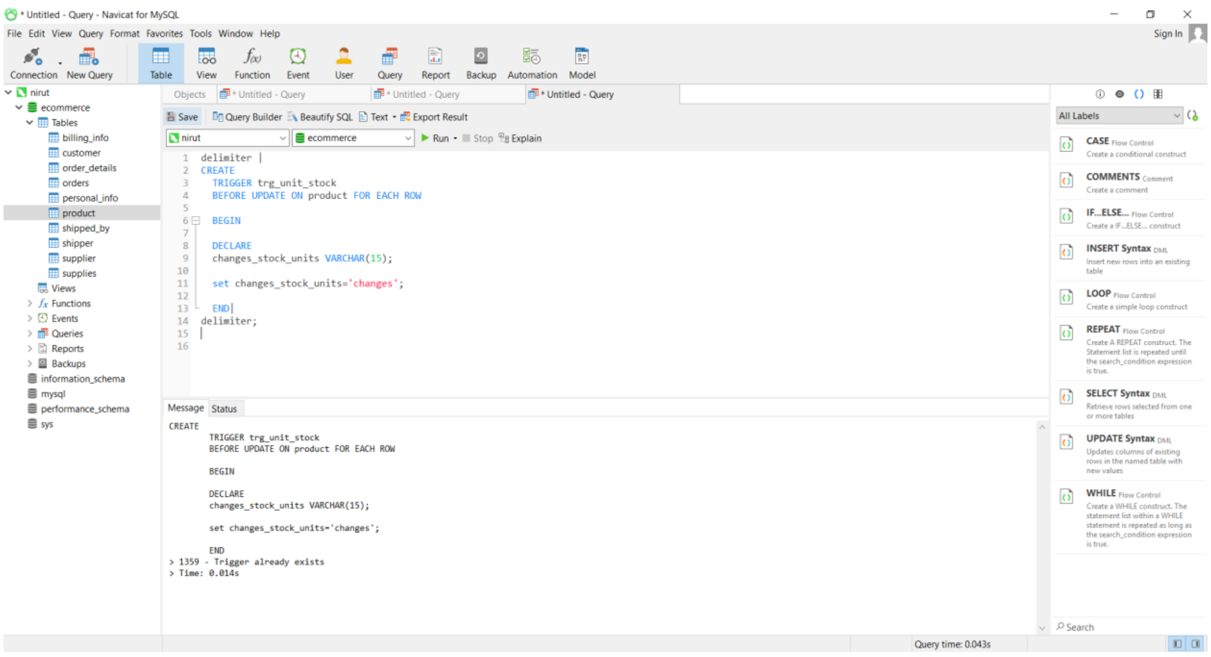
```
delimiter //
create procedure orderdesc (in id varchar(20))
begin
select orders.customer_id, customer.name, shipped_date, billing_address from orders,billing_info,customer
where
billing_info.billing_id = orders.billing_id
and
customer.customer_id = orders.customer_id
and
id = orders.customer_id;
end //
```

```
create procedure orderdesc (in id varchar(20))
begin
select orders.customer_id, customer.name, shipped_date, billing_address from orders,billing_info,customer
where
billing_info.billing_id = orders.billing_id
and
customer.customer_id = orders.customer_id
and
id = orders.customer_id;
end
> OK
> Time: 0.001s
```

Query time: 0.057s

QUERY 3:

```
delimiter |
CREATE
        TRIGGER trg_unit_stock
        BEFORE UPDATE ON product FOR EACH ROW

        BEGIN

        DECLARE
        changes_stock_units VARCHAR(15);

        set changes_stock_units='changes';

        END|
delimiter;
```



```
delimiter |
CREATE
  TRIGGER trg_unit_stock
  BEFORE UPDATE ON product FOR EACH ROW

  BEGIN

  DECLARE
  changes_stock_units VARCHAR(15);

  set changes_stock_units='changes';

  END|
delimiter;
```

```
CREATE
        TRIGGER trg_unit_stock
        BEFORE UPDATE ON product FOR EACH ROW

        BEGIN

        DECLARE
        changes_stock_units VARCHAR(15);

        set changes_stock_units='changes';

        END
> 1359 - Trigger already exists
> Time: 0.014s
```

Query time: 0.043s

CONCLUSION AND FUTURE WORK

For the entrepreneur, electronic shopping generates new business opportunities and for the customer, it makes comparative shopping possible.

Making this project was a really good learning experience; we got the feel for the basic development of an E-Commerce Database and can now appreciate the amount of work involved in making a full-fledged E-Commerce

Furthering we will try to integrate our database with Front-End and making it more pleasing, adding more features as well as incorporating a secure payment portal.