Database design (CBD 2303_4)

Design and Implementation of E-Commerce Site for Online Shopping

Shopdrop

Submitted by:

Bhupinder Kaur (C0764492)

Martha(C0761227)

Sahar Aghababaei(C0)

Manu Anmol Singh(C0773611)

Brahm Aggarwal (C0770538)

Table of contents

Requirement analysis

Company Overview:

In this project we designed and implemented an E-commerce site for online shopping named as **Shopdrop.**

- **Type of company: Retailer**

.

E-commerce is fast gaining ground as an accepted and used business paradigm. More and more business houses are implementing web sites providing functionality for performing commercial transactions over the web. It is reasonable to say that the process of shopping on the web is becoming common place.

The objective of this project is to develop a general purpose e-commerce store where traditional product from different countries can be bought from the comfort of home through the Internet.

Shopdrop is a lifestyle e-commerce web application(an online retail store) which retail various traditional products that are related to different culture and nations in different countries.This project allows viewing various products available enables registered users to purchase products , instantly using online payment and also can place orders by using Cash on Delivery option.Furthermore, every client of it have their own membership card which receive offer in different ways such as discount on buying, receiving Gift cards and etc.

Description:

- Any member can register and view available products.
- Only registered member can purchase multiple products regardless of quantity.
- Users can purchase products instantly by online payments.
- Orders can also be placed by cash on delivery option.
- Membership card is a discount and gift policy provided by company i.e. if customers have their own membership card they can receive some discounts.
- After this, order will be connected to the supplier because this is a online shopping site for traditional product from all countries so we have suppliers from different nations who supplies the product when order is confirmed.
- Then supplier send the selected product by shipping method and order will be delivered to the customer.

Product:

Our company serve traditional product from different countries.

Challenges we faced

- Defining relation of each table.
- Cardinality to the tables.
- Creating Entity relationship diagram for database.

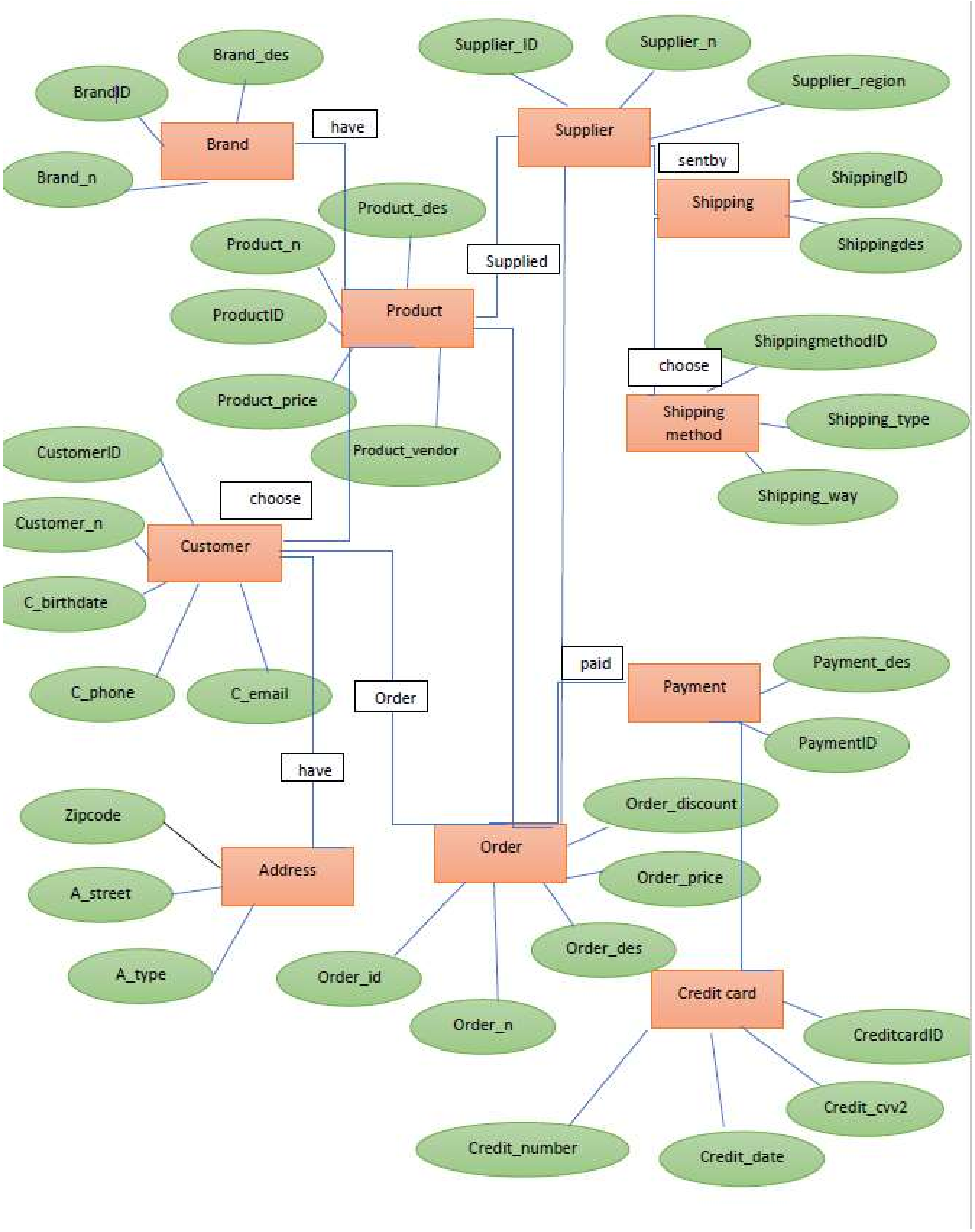
Tools used

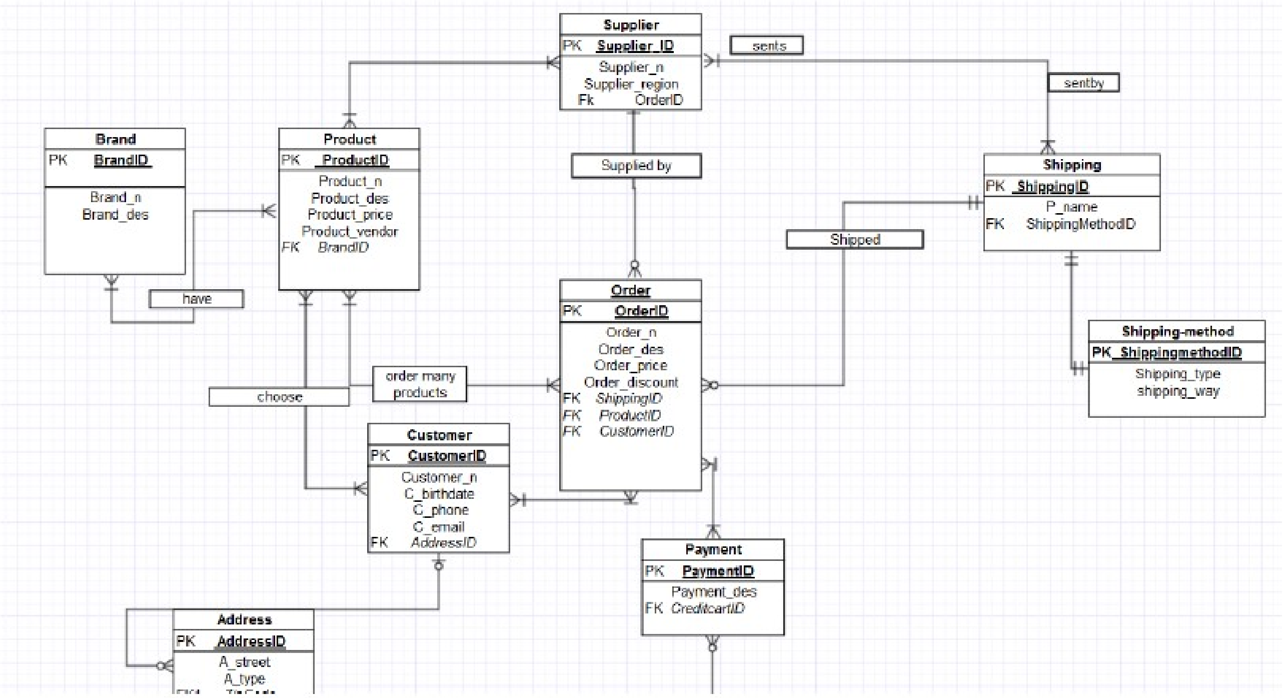
- Gliffy
- Microsoft SQL

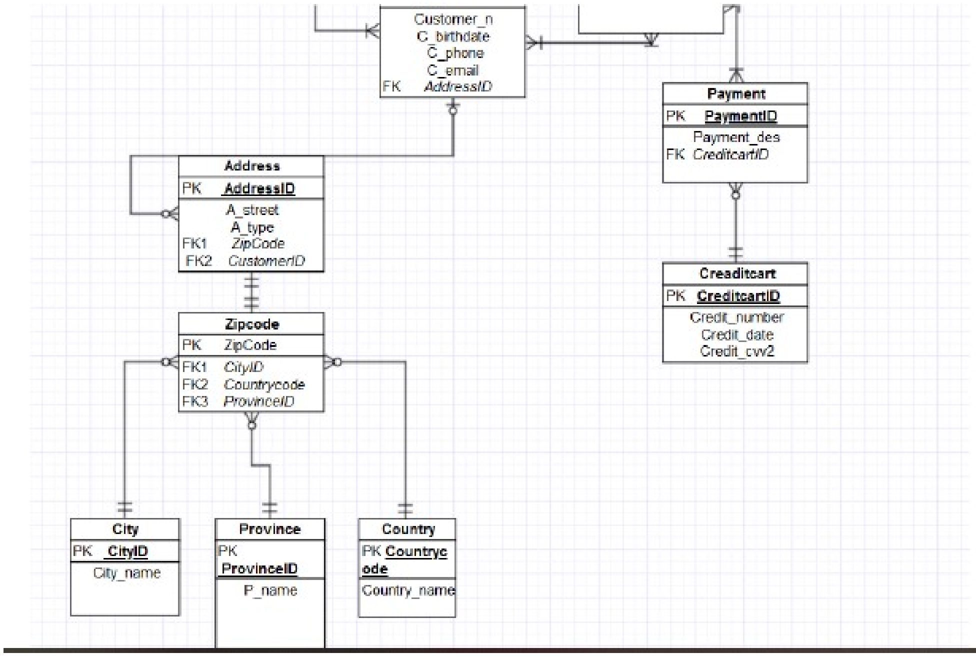
➢ **Entities and attribute Identification:** We have 18 entities and their corresponding attribures in our project given as below:

- BRAND:**Brand_ID**, Brand_n , Brand_des
- PRODUCT: **Procduct_ID** , Product_n , Product_des,Product_price,Product_vendor, Product_availibility
- PRODUCT_BRAND:**Product_ID Brand_ID** ,Attribute
- SUPPLIER: **Supplier_ID**,Supplier_region
- SHIPPING: **Shipping_ID** , P_name, ShippingMethodID
- SUPPLIER_SHIPPING:**Supplier_ShippingID,***Supplier_ID,Order_Product_ID*
- SHIPPING_METHOD:**ShippingmethodID**, Sh_air, Sh_sea, Sh_train
- ORDER: **Order_ID**, Order-n, Order_des,Order_price,Order_discount,*ShippingID*
- ORDER_PRODUCT:**Order_productID**,*ProductID,OrderID*
- CUSTOMER:**Customer_ID**,Customer_n,C_birthdate,C_phone,C_email,*Address_ID*
- CUSTOMER_ORDER:**Customer_orderID**,Attribute
- PAYMENT:**PaymenT_ID**,Payment_des,*CreditcardID*
- CREDITCARD:**CreditcardID**,Credit_number,Credit_date,Credit_cvv2
- ADDRESS:**AddressID**,A_street,A_type, ZipCode,*Customer_ID*
- ZIPCODE:Zipcode,*CityID,Countrycode, ProvinceID*
- CITY:**CityID**,City_name
- PROVINCE:**Province_ID**,P_name
- COUNTRY:**Countrycode**, Country_name

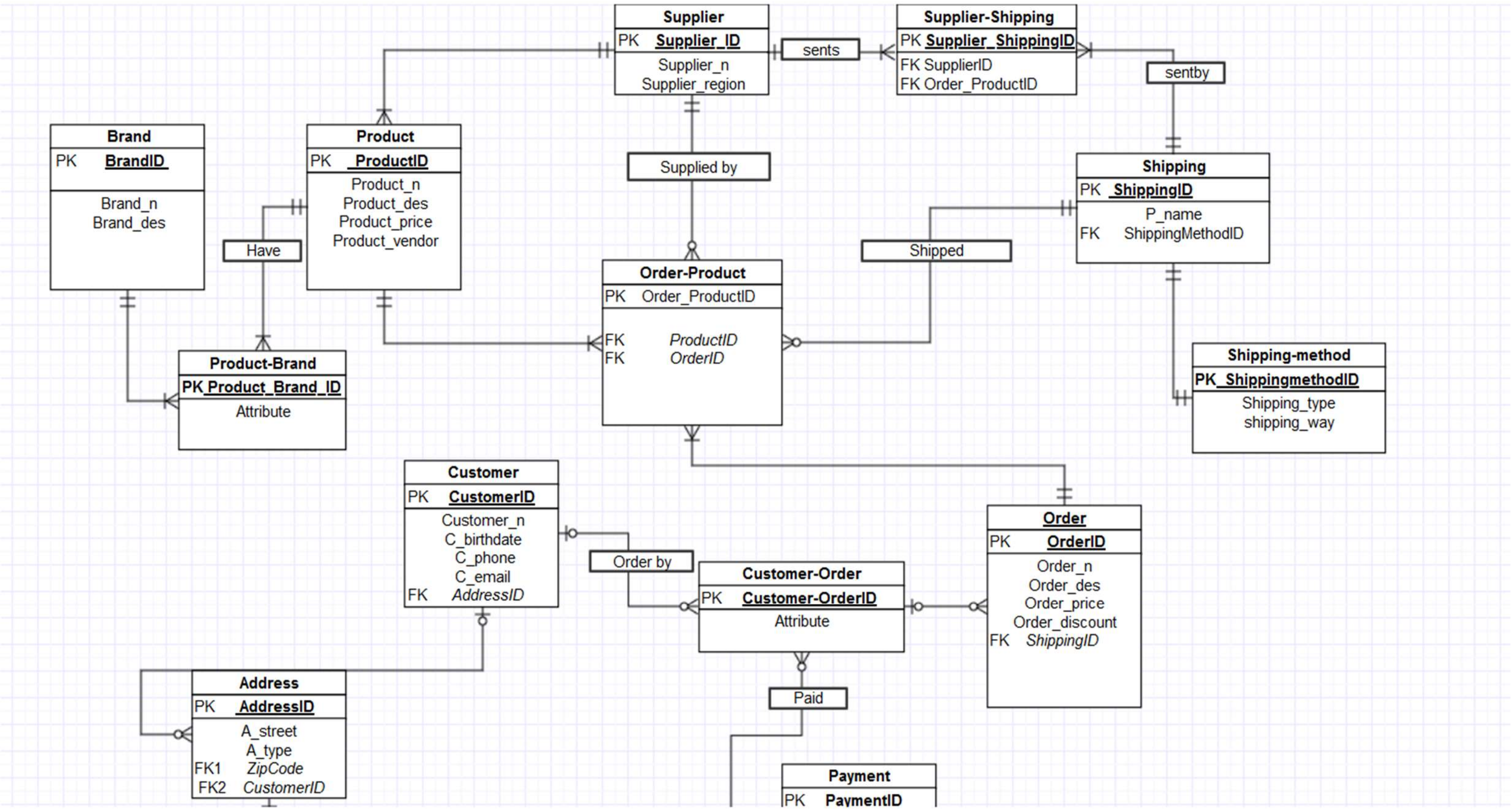
- Before resolving many to many relationship

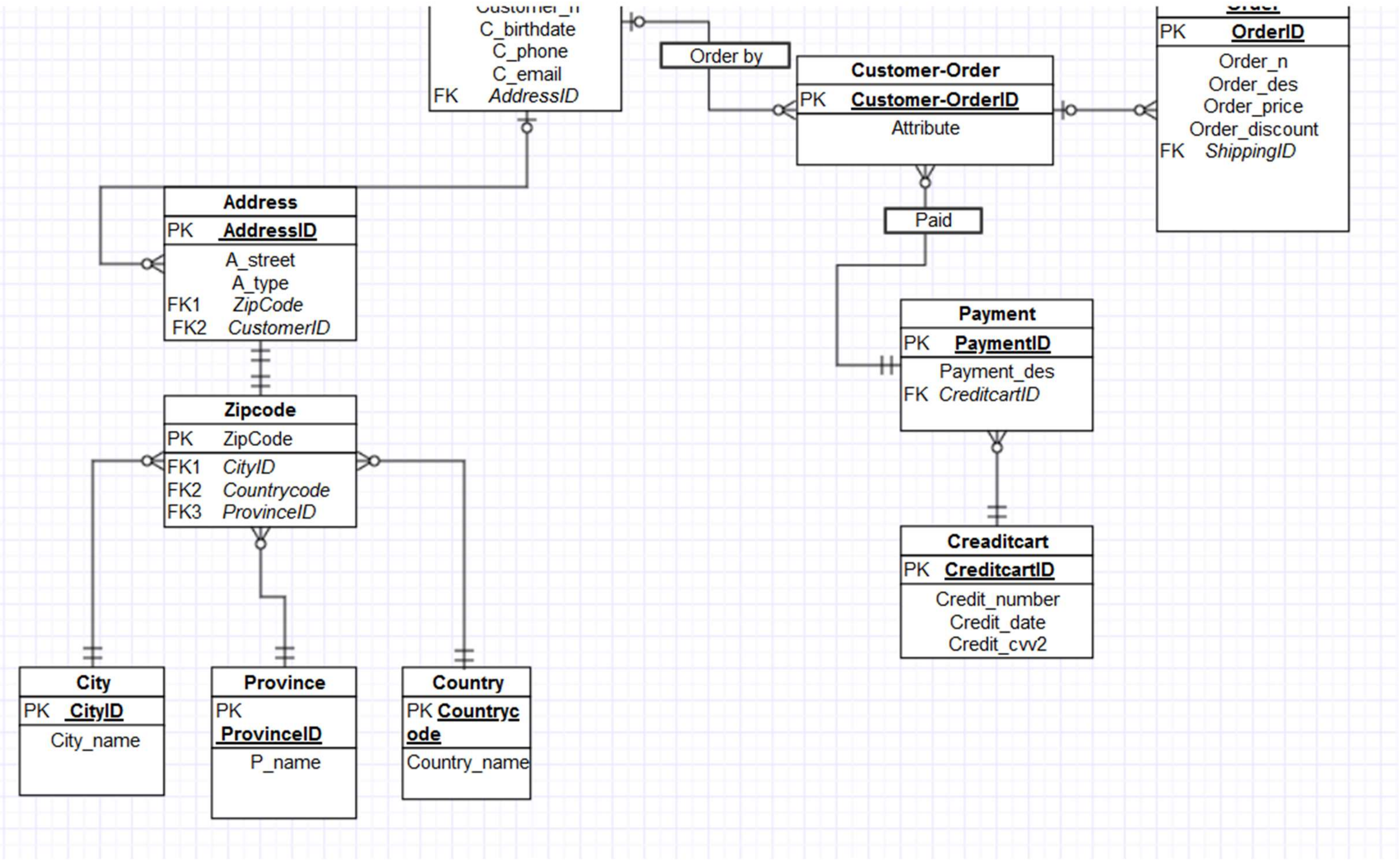



Supplier
PK Supplier_ID
Supplier_n
Supplier_region
Fk OrderID
sents
sentby
Brand
PK BrandID
Brand_n
Brand_des
Product
PK ProductID
Product_n
Product_des
Product_price
Product_vendor
FK BrandID
Supplied by
Shipping
PK ShippingID
P_name
FK ShippingMethodID
Shipped
have
Order
PK OrderID
Order_n
Order_des
Order_price
Order_discount
FK ShippingID
FK ProductID
FK CustomerID
Shipping-method
PK ShippingmethodID
Shipping_type
shipping_way
choose
order many products
Customer
PK CustomerID
Customer_n
C_birthdate
C_phone
C_email
FK AddressID
Payment
PK PaymentID
Payment_des
FK CreditcartID
Address
PK AddressID
A_street
A_type


Customer_n
C_birthdate
C_phone
C_email
FK AddressID
Payment
PK PaymentID
Payment_des
FK CreditcartID
Address
PK AddressID
A_street
A_type
FK1 ZipCode
FK2 CustomerID
Creaditcart
PK CreditcartID
Credit_number
Credit_date
Credit_cvv2
Zipcode
PK ZipCode
FK1 CityID
FK2 Countrycode
FK3 ProvinceID
City
PK CityID
City_name
Province
PK ProvinceID
P_name
Country
PK Countrycode
Country_name

ER Diagram : After Solving many to many relationship

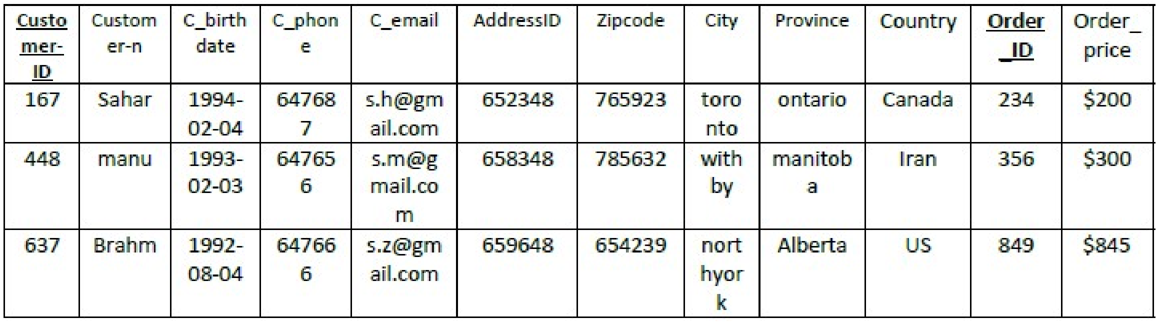



Customer_n
C_birthdate
C_phone
C_email
FK AddressID
Order by
Customer-Order
PK Customer-OrderID
Attribute
Order
PK OrderID
Order_n
Order_des
Order_price
Order_discount
FK ShippingID
Paid
Address
PK AddressID
A_street
A_type
FK1 ZipCode
FK2 CustomerID
Payment
PK PaymentID
Payment_des
FK CreditcartID
Zipcode
PK ZipCode
FK1 CityID
FK2 Countrycode
FK3 ProvinceID
Creaditcart
PK CreditcartID
Credit_number
Credit_date
Credit_cvv2
City
PK CityID
City_name
Province
PK ProvinceID
P_name
Country
PK Countrycode
Country_name

Logical Model:

Normalization

Customer

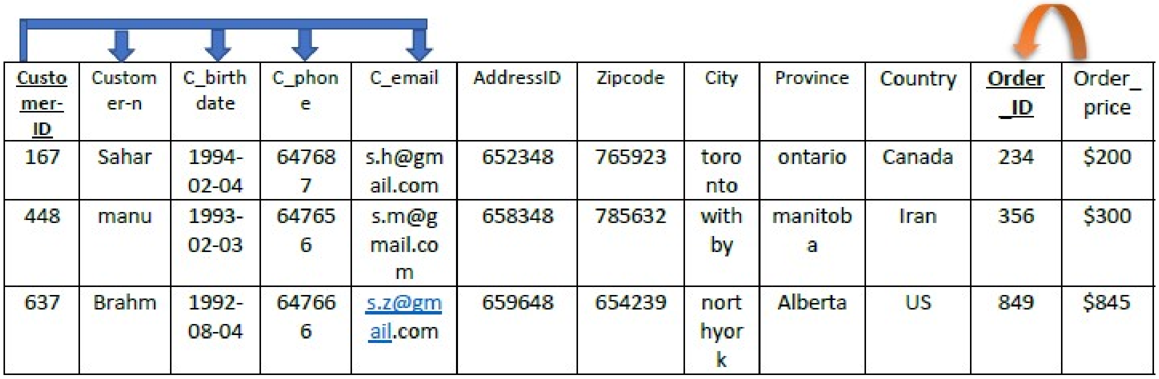


Customer-ID	Customer-n	C_birthdate	C_phone	C_email	AddressID	Zipcode	City	Province	Country	Order_ID	Order_price
167	Sahar	1994-02-04	647687	s.h@gmail.com	652348	765923	toronto	ontario	Canada	234	$200
448	manu	1993-02-03	647656	s.m@gmail.com	658348	785632	withby	manitoba	Iran	356	$300
637	Brahm	1992-08-04	647666	s.z@gmail.com	659648	654239	northyork	Alberta	US	849	$845

*It is in 1NF and each attributes contains a single value.

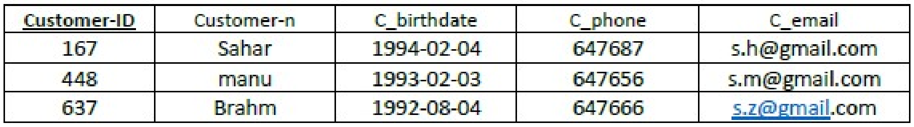
*The primary key is a composite key consisting of Customer_ID and Order_ID.

Now it has all of conditions to convert it in the 2NF:



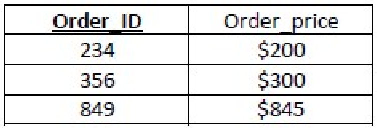
Customer-ID	Customer-n	C_birthdate	C_phone	C_email	AddressID	Zipcode	City	Province	Country	Order_ID	Order_price
167	Sahar	1994-02-04	647687	s.h@gmail.com	652348	765923	toronto	ontario	Canada	234	$200
448	manu	1993-02-03	647656	s.m@gmail.com	658348	785632	withby	manitoba	Iran	356	$300
637	Brahm	1992-08-04	647666	s.z@gmail.com	659648	654239	northyork	Alberta	US	849	$845

Customer



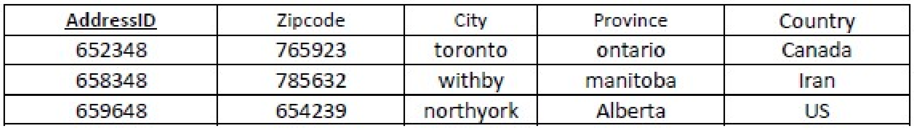
Customer-ID	Customer-n	C_birthdate	C_phone	C_email
167	Sahar	1994-02-04	647687	s.h@gmail.com
448	manu	1993-02-03	647656	s.m@gmail.com
637	Brahm	1992-08-04	647666	s.z@gmail.com

Order



Order_ID	Order_price
234	$200
356	$300
849	$845

Address

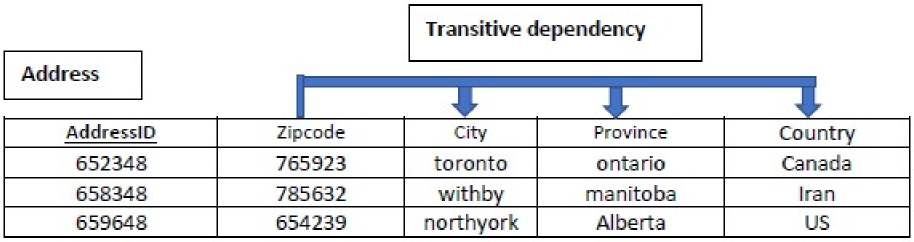


AddressID	Zipcode	City	Province	Country
652348	765923	toronto	ontario	Canada
658348	785632	withby	manitoba	Iran
659648	654239	northyork	Alberta	US

* It is in 2NF and there are no partial key dependencies.

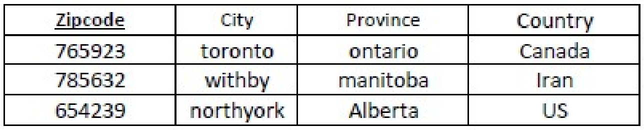
*The primary key in tables Customer and Order become a foreign key in the Main table.

Now it has all of conditions to convert it in the 3NF:

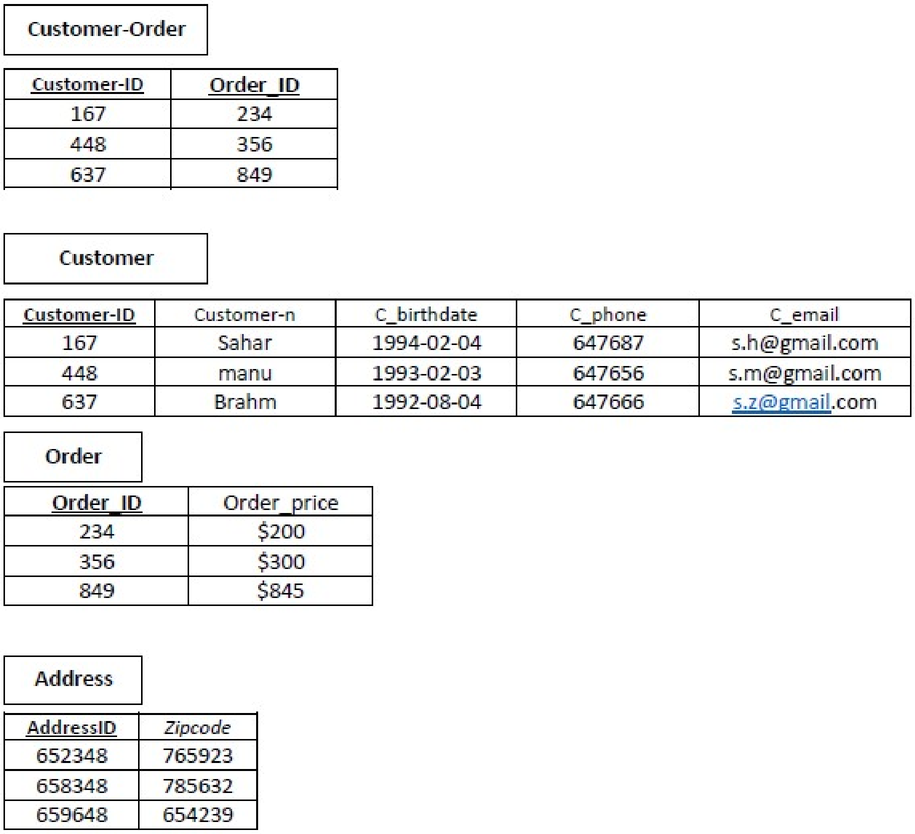


AddressID	Zipcode	City	Province	Country
652348	765923	toronto	ontario	Canada
658348	785632	withby	manitoba	Iran
659648	654239	northyork	Alberta	US

Zipcode



Zipcode	City	Province	Country
765923	toronto	ontario	Canada
785632	withby	manitoba	Iran
654239	northyork	Alberta	US



Customer-Order

Customer-ID	Order_ID
167	234
448	356
637	849

Customer

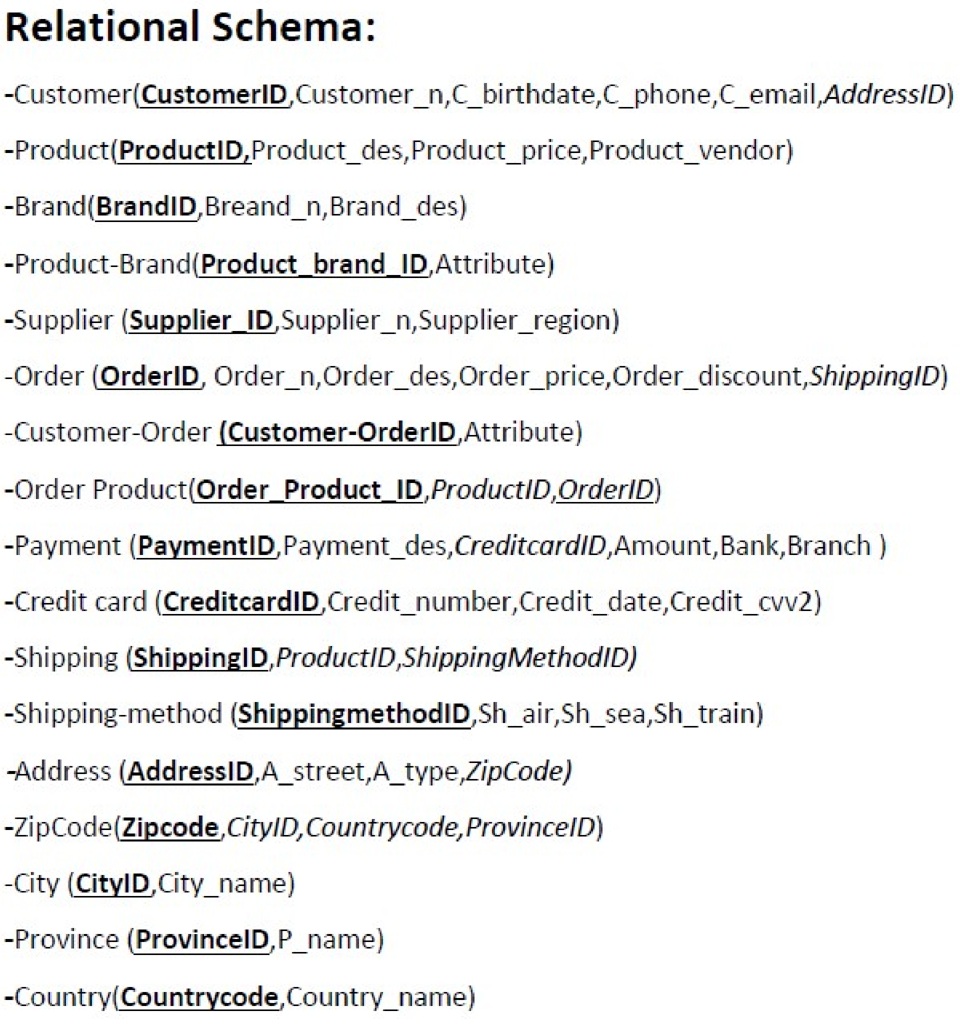
Customer-ID	Customer-n	C_birthdate	C_phone	C_email
167	Sahar	1994-02-04	647687	s.h@gmail.com
448	manu	1993-02-03	647656	s.m@gmail.com
637	Brahm	1992-08-04	647666	s.z@gmail.com

Order

Order_ID	Order_price
234	$200
356	$300
849	$845

Address

AddressID	*Zipcode*
652348	765923
658348	785632
659648	654239



Relational Schema:

-Customer(**CustomerID**,Customer_n,C_birthdate,C_phone,C_email,*AddressID*)

-Product(**ProductID,**Product_des,Product_price,Product_vendor)

-Brand(**BrandID**,Breand_n,Brand_des)

-Product-Brand(**Product_brand_ID**,Attribute)

-Supplier (**Supplier_ID**,Supplier_n,Supplier_region)

-Order (**OrderID**, Order_n,Order_des,Order_price,Order_discount,*ShippingID*)

-Customer-Order **(Customer-OrderID**,Attribute)

-Order Product(**Order_Product_ID**,*ProductID*,*OrderID*)

-Payment (**PaymentID**,Payment_des,*CreditcardID*,Amount,Bank,Branch)

-Credit card (**CreditcardID**,Credit_number,Credit_date,Credit_cvv2)

-Shipping (**ShippingID**,*ProductID*,*ShippingMethodID)*

-Shipping-method (**ShippingmethodID**,Sh_air,Sh_sea,Sh_train)

-Address (**AddressID**,A_street,A_type,*ZipCode)*

-ZipCode(**Zipcode**,*CityID,Countrycode,ProvinceID*)

-City (**CityID**,City_name)

-Province (**ProvinceID**,P_name)

-Country(**Countrycode**,Country_name)

Sql Query :

```
create database classPro
use classPro

create table brand (
brandID numeric(4,0) NOT NULL,
brand_n integer not null,
brand_desc character(20) not null,
CONSTRAINT brand_brandID_pk PRIMARY KEY (brandID));  /* CONS PRIMARY KEY */

select * from brand
create table product (
product_id numeric(4,0) NOT NULL,
product_name character(25) not null,
product_desc character(20) not null,
product_price numeric(4,0) not null,
product_vendor character (20) not null,
product_availability character (3) not null,
CONSTRAINT product_product_id_pk PRIMARY KEY (product_id), /* CONS PRIMARY KEY*/
CONSTRAINT product_product_availability_ck CHECK (product_availability IN('YES','NO')) /* CONS CHECK YES,NO*/
);



create table product_brand(
brandID numeric(4,0),
product_id numeric(4,0),
offer character(40),
CONSTRAINT product_brand_brand_id_pk PRIMARY KEY (brandID),
constraint product_brand_fk foreign key(brandID) references brand(brandID),
constraint product_brand_fk1 foreign key(product_id) references product(product_id));


select * from product_brand;

create table supplier(
supplier_id numeric(4,0) NOT NULL,
supplier_name character(25) not null,
supplier_region character(20) not null,
CONSTRAINT supplier_supplier_id_pk PRIMARY KEY (supplier_id),
CONSTRAINT supplier_supplier_region_ck CHECK (supplier_region IN('ON','QC')) /* CONS CHECK region*/
);

create table supplier_shipping (
supplier_shippingid numeric(4,0) NOT NULL,
supplier_id numeric(4,0) NOT NULL,
CONSTRAINT supplier_shipping_id_supplier_shippingid_pk PRIMARY KEY (supplier_shippingid),
FOREIGN KEY(supplier_id) REFERENCES supplier(supplier_id)
);

create table shipping_method(
shippingmethodid numeric (4,0) NOT NULL,
shipping_type character(20) not null,
shipping_way character (20) not null,
CONSTRAINT shipping_method_shippingmethodid_pk PRIMARY KEY (shippingmethodid),
CONSTRAINT shipping_method_shipping_way_ck CHECK (shipping_way IN('sea','air','train')), /* CONS CHECK way of
shipping*/
CONSTRAINT shipping_method_shipping_type_ck CHECK (shipping_type IN('standard','express')) /* CONS CHECK type of
shipping*/
);

create table shipping (
shipping_id numeric(4,0) NOT NULL,
P_name character (20) NOT NULL,
shippingmethodid numeric (4,0) NOT NULL,
CONSTRAINT shipping_shipping_id_pk PRIMARY KEY (shipping_id), /* CONS primary key */
```

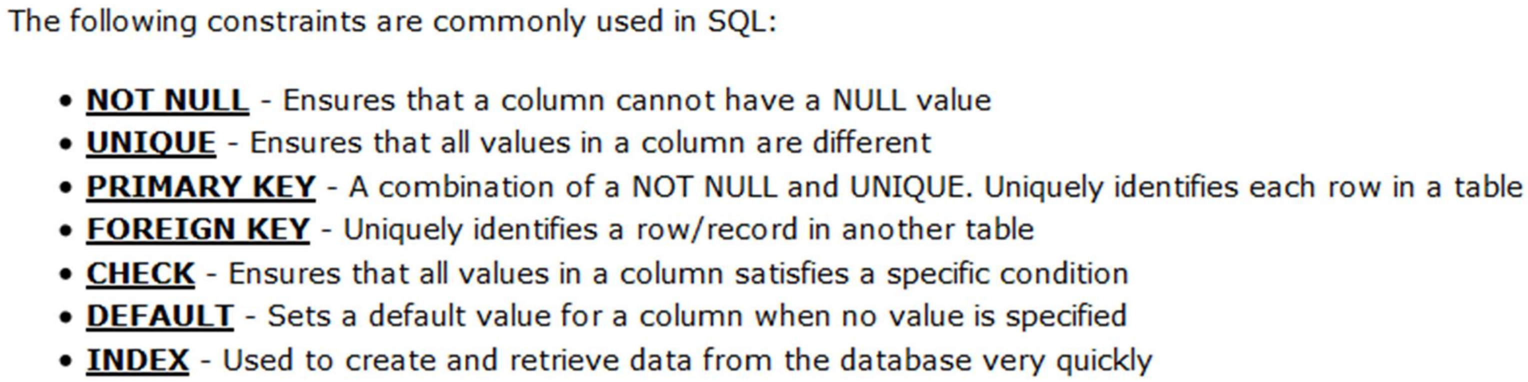
```
FOREIGN KEY(shippingmethodid) REFERENCES shipping_method(shippingmethodid)
);

create table order1(
order_id numeric(4,0) NOT NULL,
order_des character (20) NOT NULL,
order_price numeric(4,0) NOT NULL,
order_discount numeric(2,0) NOT NULL ,
shipping_id numeric(4,0) NOT NULL,
CONSTRAINT order1_order_id_pk PRIMARY KEY (order_id), /* CONS primary key */
FOREIGN KEY(shipping_id) REFERENCES shipping(shipping_id),
CONSTRAINT order1_order_discount_ck CHECK(order_discount<100 AND ORDER_PRICE>100)  /* CONS CHECK discount
*/
);

ALTER TABLE order1  /* CONS CHECK price of order */
        ADD CONSTRAINT order1_order_price_ck CHECK(order_price >= 1);

create table order_product(
order_productid numeric(4,0) NOT NULL
CONSTRAINT order_product_order_productid_pk PRIMARY KEY (order_productid),
product_id numeric(4,0) NOT NULL,
order_id numeric(4,0) NOT NULL,
constraint order_product_fk foreign key(product_id) references product(product_id),
constraint order_product_fk1 foreign key(order_id)  references order1(order_id));



create table customer(
customer_id numeric(4,0) not null,
customer_name character(25) not null,
customer_bday date not null,
customer_phone numeric(10,0) not null,
customer_email character(30) not null,
CONSTRAINT customer_customer_id_pk PRIMARY KEY (customer_id),
CONSTRAINT customer_customer_phone_uq UNIQUE(customer_phone),  /*CONS UNIQUE PHONE NO. */
CONSTRAINT customer_customer_phone_ck CHECK(len(customer_phone)=10),  /*CONS CHECK LENGTH */
CONSTRAINT customer_customer_email_ck CHECK(customer_email like '%_@__%.__%')); /* CONS CHECK : EMAIL
FORMAT*/

ALTER TABLE customer /* CONS CHECK DATE < '2009-01-01' */
  ADD CONSTRAINT customer_customer_bday_ck
    CHECK(customer_bday < '2009-01-01');

create table customer_order(
customer_id numeric(4,0) not null primary key,
)

create table province(
provinceID numeric (4,0) not null,
p_name character (25),
CONSTRAINT province_provinceID_pk PRIMARY KEY (provinceID));
```

```
create table city(
cityID numeric (4,0) not null,
city_name character (25) NOT NULL,
CONSTRAINT city_cityID_pk PRIMARY KEY (cityID)
);
ALTER TABLE city /* CONS deafult CITY TABLE*/
        ADD CONSTRAINT city_city_name_df
        DEFAULT 'TORONTO' FOR city_name;

create table country(
countryCode numeric (4,0) not null,
country_name character (25),
CONSTRAINT country_countryCode_pk PRIMARY KEY (countryCode)
);

create table zipcode(
zipCode numeric(10,0) not null,
cityID numeric (4,0) not null,
provinceID numeric (4,0) not null,
countryCode numeric (4,0) not null,
CONSTRAINT zipcode_zipCode_pk PRIMARY KEY (zipCode),
FOREIGN KEY(cityID) REFERENCES city(cityID),
FOREIGN KEY(provinceID) REFERENCES province(provinceID),
FOREIGN KEY(countryCode) REFERENCES country(countryCode));


create table hAddress(
hAddressID numeric(4,0)not null,
a_street character(25) not null,
a_type character (25) not null,
zipCode numeric(10,0) not null,
customer_id numeric(4,0) not null,
CONSTRAINT hAddress_hAddressID_pk PRIMARY KEY (hAddressID),
FOREIGN KEY(zipCode) REFERENCES zipcode(zipCode),
FOREIGN KEY(customer_id) REFERENCES customer(customer_id));


create table creditcard(
creditcardID numeric(15,0) NOT NULL,
credit_number varchar(6) NOT NULL,
credit_date DATE NOT NULL,
credit_cvv2 NUMERIC(4,0) NOT NULL,
CONSTRAINT creditcard_creditcardID_pk PRIMARY KEY (creditcardID)
);

ALTER TABLE creditcard  /* CONS CHECK len(credit numer)=6 */
        ADD CONSTRAINT creditcard_credit_number_ck CHECK(len(credit_number)=6);

ALTER TABLE creditcard /*CONS UNIQUE (credit_number)*/
        ADD CONSTRAINT creditcard_credit_number_uq UNIQUE(credit_number);

ALTER TABLE creditcard  /* CONS CHECK len(cvv number)=3 */
        ADD CONSTRAINT creditcard_credit_cvv2_ck CHECK(len(credit_cvv2)=3);
```

```
ALTER TABLE creditcard /* CONS CHECK UNIQUE */
            ADD CONSTRAINT creditcard_credit_cvv2_uq UNIQUE(credit_cvv2);

create table payment(
paymentID varchar(20) NOT NULL,
payment_des varchar(100),
creditcardID numeric(15,0) NOT NULL,
CONSTRAINT payment_paymentID_pk PRIMARY KEY (paymentID),
FOREIGN KEY(creditcardID) REFERENCES creditcard(creditcardID));

select * from sysobjects where xtype='U';
select * from brand ;
select * from product;
select * from product_brand;
select * from supplier;
select * from supplier_shipping;
select * from shipping_method;
select * from shipping;
select * from order1;
select * from order_product;
select * from customer;
select * from customer_order;
select * from province;
select * from city;
select * from country;
select * from zipcode;
select * from hAddress;
select * from creditcard;
select * from payment;




INSERT INTO brand VALUES('0001','10','Samsung'),
('0002','20','Apple'),
('0003','30','HP'),
('0004','40','Dell'),
('0005','50','Comapq'),
('0006','60','Microsoft'),
('0007','70','Seagate'),
('0008','80','Transcend'),
('0009','90','LG'),
('0010','100','OnePlus');



INSERT INTO product VALUES ('101','mobile','S9','500','rajesh','yes'),
 ('102','computer','ab032tx','600','hariom','yes'),
 ('103','mouse','wireless','10','kevin','no'),
 ('104','mobile','s10','1000','jeevan','no'),
 ('105','notebook','chrome','400','awesome','yes'),
 ('106','laptop','ab012','650','richard','no');
 INSERT INTO product VALUES  ('107','macbook','pro256','999','awesome','');


INSERT INTO product_brand VALUES('0001','101','free headphones'),
('0002','102','discount'),
('0003','103','free battery'),
('0004','104','free insurance'),
('0005','105','free bag'),
('0006','106','nothing free');


INSERT INTO supplier VALUES('4001','Balla','ON'),
('4002','lalu','ON'),
('4003','jane','ON'),
('4004','jenny','ON'),
('4005','tim','QC'),
('4006','kapoor','QC'),
('4007','jack','QC'),
('4008','queenS','ON');
INSERT INTO supplier VALUES('4009','vicky','bc');


INSERT INTO supplier_shipping  VALUES('1001','4001'),
('1002','4002'),
('1003','4003'),
('1004','4004'),
('1005','4005'),
('1006','4006'),
('1007','4007');



INSERT INTO shipping_method VALUES('3001','standard','sea'),
 ('3002','standard','sea'),
 ('3003','express','train'),
 ('3004','standard','sea'),
 ('3005','standard','train'),
 ('3006','express','air'),
 ('3007','standard','train'),
 ('3008','express','air'),
 ('3009','standard','train'),
 ('3010','express','air'),
 ('3011','express','air');
 INSERT INTO shipping_method VALUES('3012','standard','bus');




INSERT INTO shipping  VALUES ('5001', 'william', '3001'),
('5002', 'roony', '3002'),
('5003', 'avi', '3003'),
('5004', 'rommy', '3004'),
('5005', 'dkbhose', '3005'),
('5006', 'ravi', '3006'),
('5007', 'kavi', '3307'),
('5008', 'alex', '3008'),
('5009', 'shane', '3009'),
('5010', 'kim', '3010');

select * from shipping_method



INSERT INTO order1 VALUES('6001','iphone','450','20','5001'),
('6002','macbook','900','10','5002'),
('6003','chrome','675','30','5003'),
('6004','mouse','30','10','5004'),
('6005','ipad','850','20','5005'),
('6006','macbookPro','1000','10','5006'),
('6007','hp','599','10','5007'),
('6008','samsung s8','499','50','5008'),
('6009','tablet','299','60','5009'),
('6010','OLED samsung','999','5','50010');
INSERT INTO order1 VALUES('6011','iphone','0','5','5011');

INSERT INTO order_product VALUES('7001','101','6001'),
('7002','102','6002'),
('7003','103','6003'),
('7004','104','6004'),
('7005','105','6005'),
('7006','106','6006'),
('7007','107','6007');


INSERT INTO customer VALUES('8001','sri','2008-01-01',9888823257,'sri@gmail.com'),
 ('8002','ak','2007-01-01',9888823251,'ak@gmail.com'),
 ('8003','bk','2006-01-01',9888823252,'bk@gmail.com'),
 ('8004','ck','2005-01-01',9888823253,'ck@gmail.com'),
 ('8005','dk','2004-01-01',9888823254,'dk@gmail.com'),
 ('8006','ek','2002-01-01',9888823255,'ek@gmail.com'),
 ('8007','gk','2003-01-01',9888823299,'gk@gmail.com'),
 ('8008','hk','2001-01-01',9888823258,'hk@gmail.com'),
 ('8009','ik','1999-01-01',9888823259,'ik@gmail.com'),
 ('8010','kk','1998-01-01',9888823210,'kk@gmail.com');
 INSERT INTO customer VALUES('8011','jk','2008-01-01',9888823257,'jk@gmail.com');
 INSERT INTO customer VALUES('8012','lk','2010-01-01',9888823219,'sri@gmail.com');
 INSERT INTO customer VALUES('8013','mk','2008-01-01',988882328,'mk@gmail.com');
  INSERT INTO customer VALUES('8014','nk','2008-01-01',9888823278,'nk.gmail.com');




INSERT INTO customer_order VALUES('9001'),
('9002'),
('9003'),
('9004'),
('9005'),
('9006'),
('9007'),
('9008'),
('9009'),
('9010');



INSERT INTO province VALUES('4501','ON'),
('4502','ON'),
('4503','ON'),
('4504','BC'),
('4505','BC'),
('4506','QC'),
('4507','QC'),
('4508','BC'),
('4509','ON'),
('4510','ON');

INSERT INTO city VALUES('5501','TORONTO'),
('5502','SCARBOROUGH'),
('5503','ETOBICOK'),
('5504','BRAMPTON'),
('5505','MISSISSAUGA'),
('5506','VAUGHAN'),
('5507','OSHAWA'),
('5508','COBOURGH'),
('5509','HALTON HILLS'),
('5510','KITCHENER');
INSERT INTO city VALUES('5512','');
select* from city where cityID = 5512 ;



INSERT INTO country VALUES('6501','CANADA'),
('6502','USA'),
('6503','UK'),
('6504','INDIA'),
('6505','RUSSIA'),
('6506','IRAN');


INSERT INTO zipcode VALUES('7501','5508','4501','6501'),
('7502','5501','4501','6501'),
('7503','5502','4501','6501'),
('7504','5503','4501','6501'),
('7505','5504','4501','6501'),
('7506','5505','4501','6501'),
('7507','5506','4501','6501'),
('7508','5507','4501','6501');
```

```
INSERT INTO hAddress VALUES('8501','JANE ST.','APT','7501','9001'),
('8502','CHURCH ST.','APT','7502','9002'),
('8503','ALEXA ST.','HOUSE','7503','9003'),
('8504','RICH ST.','VILLA','7504','9004'),
('8505','BC ST.','APT','7505','9005'),
('8506','BANK ST.','BUILDING','7506','9006'),
('8507','TESLA ST.','HOUSE','7507','9007'),
('8508','BENZ ST.','VILLA','7508','9008');


INSERT INTO creditcard VALUES('112241','998871','2020-01-01','901'),
('112242','998872','2021-03-01','902'),
('112243','998873','2022-09-01','903'),
('112244','998874','2023-01-01','904'),
('112245','998875','2024-01-01','905'),
('112246','998876','2020-01-01','906'),
('112247','998877','2022-01-01','907'),
('112248','998878','2023-05-01','908'),
('112249','998879','2024-06-01','909'),
('112210','998810','2023-08-01','910'),
('112211','998811','2021-02-01','911'),
('112212','998812','2021-04-01','912');
INSERT INTO creditcard VALUES('112213','998813','2020-01-01','901');




INSERT INTO payment VALUES('P001','VISA','112241'),
('P002','VISA','112242'),
('P003','MASTERCARD','112243'),
('P004','VISA','112244'),
('P005','MASTERCARD','112245'),
('P006','MASTERCARD','112246'),
('P007','VISA','112247'),
('P008','MASTERCARD','112248'),
('P009','VISA','112249');
```

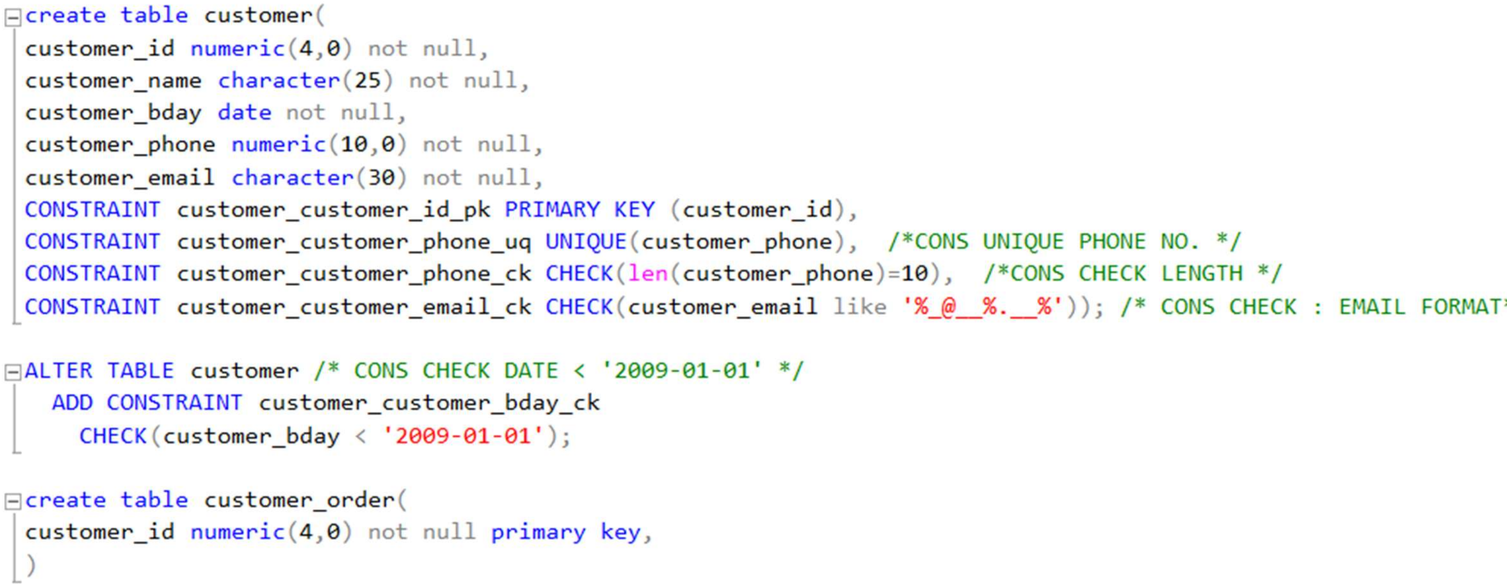
Constraints:



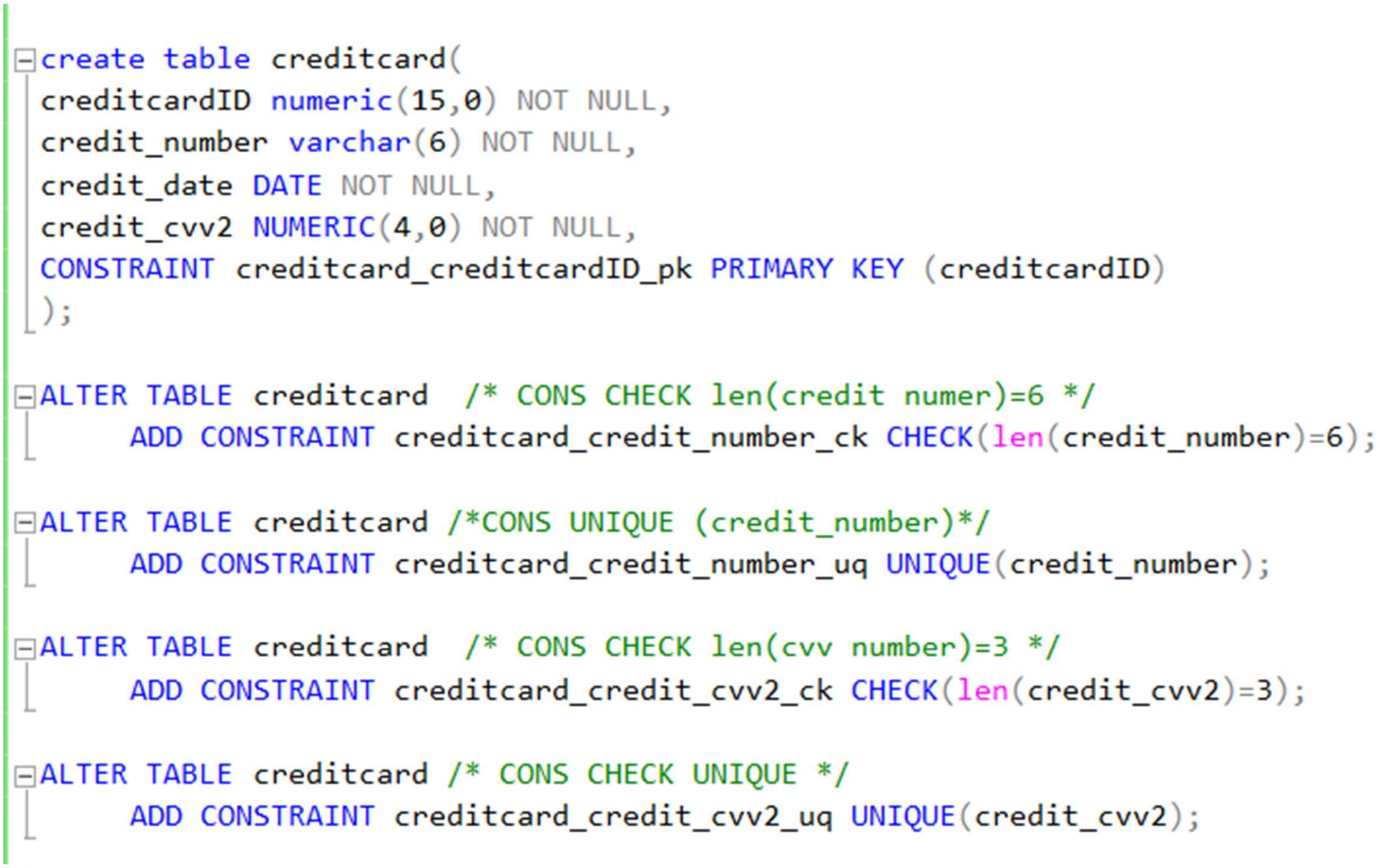
The following constraints are commonly used in SQL:

- **NOT NULL** - Ensures that a column cannot have a NULL value
- **UNIQUE** - Ensures that all values in a column are different
- **PRIMARY KEY** - A combination of a NOT NULL and UNIQUE. Uniquely identifies each row in a table
- **FOREIGN KEY** - Uniquely identifies a row/record in another table
- **CHECK** - Ensures that all values in a column satisfies a specific condition
- **DEFAULT** - Sets a default value for a column when no value is specified
- **INDEX** - Used to create and retrieve data from the database very quickly

1. Constraints :

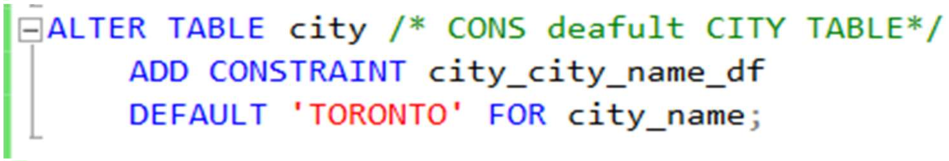


```
create table customer(
customer_id numeric(4,0) not null,
customer_name character(25) not null,
customer_bday date not null,
customer_phone numeric(10,0) not null,
customer_email character(30) not null,
CONSTRAINT customer_customer_id_pk PRIMARY KEY (customer_id),
CONSTRAINT customer_customer_phone_uq UNIQUE(customer_phone),  /*CONS UNIQUE PHONE NO. */
CONSTRAINT customer_customer_phone_ck CHECK(len(customer_phone)=10),  /*CONS CHECK LENGTH */
CONSTRAINT customer_customer_email_ck CHECK(customer_email like '%_@__%.__%')); /* CONS CHECK : EMAIL FORMAT*

ALTER TABLE customer /* CONS CHECK DATE < '2009-01-01' */
  ADD CONSTRAINT customer_customer_bday_ck
    CHECK(customer_bday < '2009-01-01');

create table customer_order(
customer_id numeric(4,0) not null primary key,
)
```



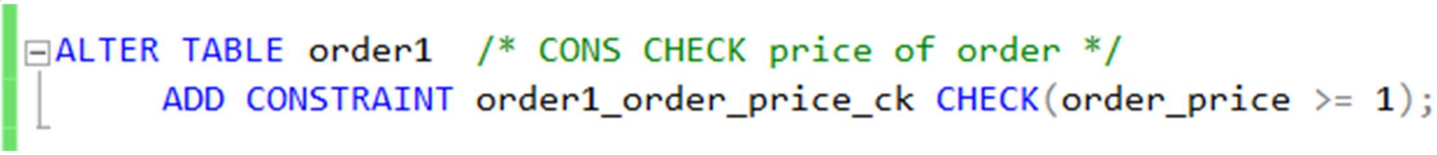
```
create table creditcard(
creditcardID numeric(15,0) NOT NULL,
credit_number varchar(6) NOT NULL,
credit_date DATE NOT NULL,
credit_cvv2 NUMERIC(4,0) NOT NULL,
CONSTRAINT creditcard_creditcardID_pk PRIMARY KEY (creditcardID)
);

ALTER TABLE creditcard  /* CONS CHECK len(credit numer)=6 */
     ADD CONSTRAINT creditcard_credit_number_ck CHECK(len(credit_number)=6);

ALTER TABLE creditcard /*CONS UNIQUE (credit_number)*/
     ADD CONSTRAINT creditcard_credit_number_uq UNIQUE(credit_number);

ALTER TABLE creditcard  /* CONS CHECK len(cvv number)=3 */
     ADD CONSTRAINT creditcard_credit_cvv2_ck CHECK(len(credit_cvv2)=3);

ALTER TABLE creditcard /* CONS CHECK UNIQUE */
     ADD CONSTRAINT creditcard_credit_cvv2_uq UNIQUE(credit_cvv2);
```

2.



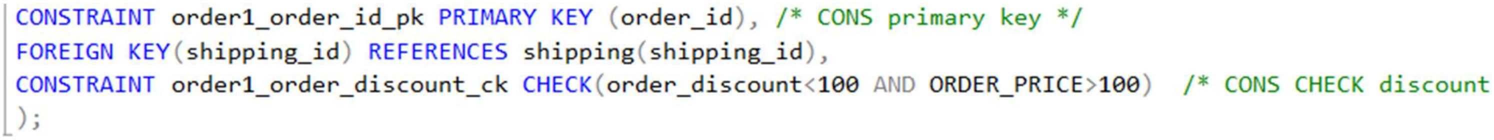
```
ALTER TABLE city /* CONS deafult CITY TABLE*/
    ADD CONSTRAINT city_city_name_df
    DEFAULT 'TORONTO' FOR city_name;
```

3.



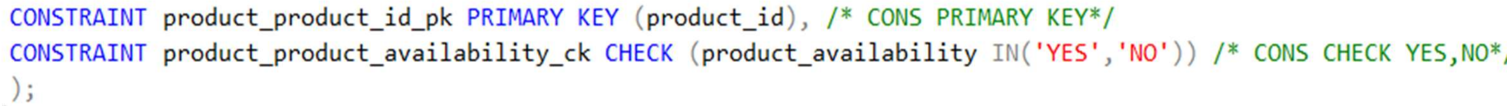
```
ALTER TABLE order1  /* CONS CHECK price of order */
     ADD CONSTRAINT order1_order_price_ck CHECK(order_price >= 1);
```

4.



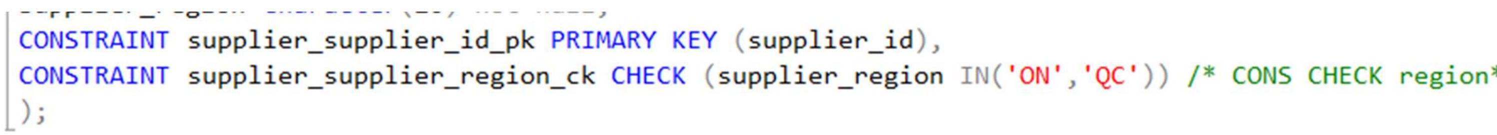
```
CONSTRAINT order1_order_id_pk PRIMARY KEY (order_id), /* CONS primary key */
FOREIGN KEY(shipping_id) REFERENCES shipping(shipping_id),
CONSTRAINT order1_order_discount_ck CHECK(order_discount<100 AND ORDER_PRICE>100)  /* CONS CHECK discount
);
```

5.



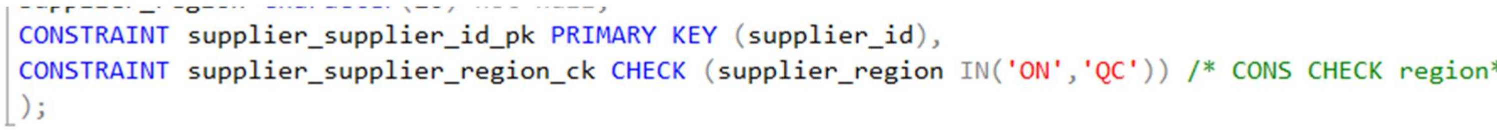
```
CONSTRAINT product_product_id_pk PRIMARY KEY (product_id), /* CONS PRIMARY KEY*/
CONSTRAINT product_product_availability_ck CHECK (product_availability IN('YES','NO')) /* CONS CHECK YES,NO*
);
```

6.



```
CONSTRAINT supplier_supplier_id_pk PRIMARY KEY (supplier_id),
CONSTRAINT supplier_supplier_region_ck CHECK (supplier_region IN('ON','QC')) /* CONS CHECK region*
);
```

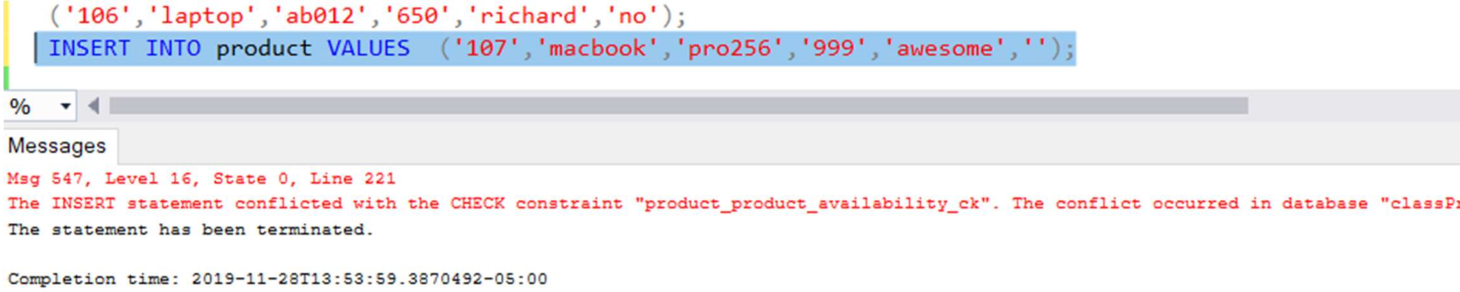
7.



```
CONSTRAINT supplier_supplier_id_pk PRIMARY KEY (supplier_id),
CONSTRAINT supplier_supplier_region_ck CHECK (supplier_region IN('ON','QC')) /* CONS CHECK region*
);
```

8.

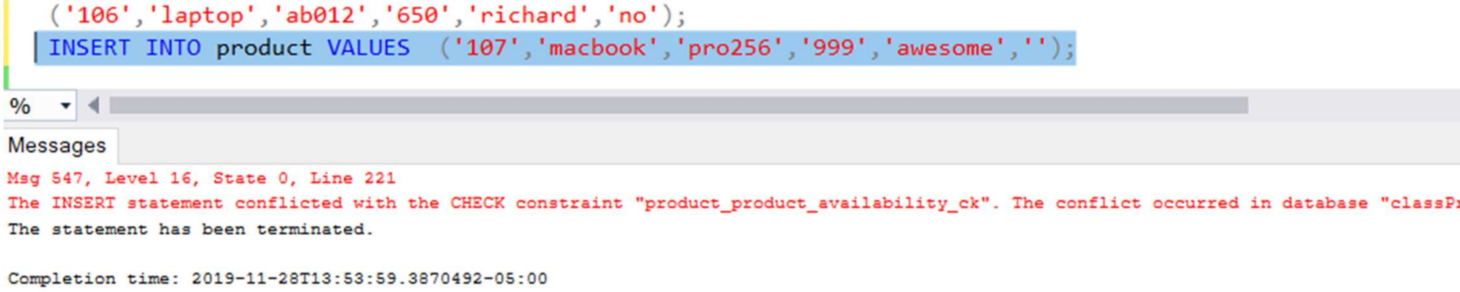
Constraints testing:



```
('106','laptop','ab012','650','richard','no');
INSERT INTO product VALUES  ('107','macbook','pro256','999','awesome','');
```

```
Msg 547, Level 16, State 0, Line 221
The INSERT statement conflicted with the CHECK constraint "product_product_availability_ck". The conflict occurred in database "classPr
The statement has been terminated.

Completion time: 2019-11-28T13:53:59.3870492-05:00
```

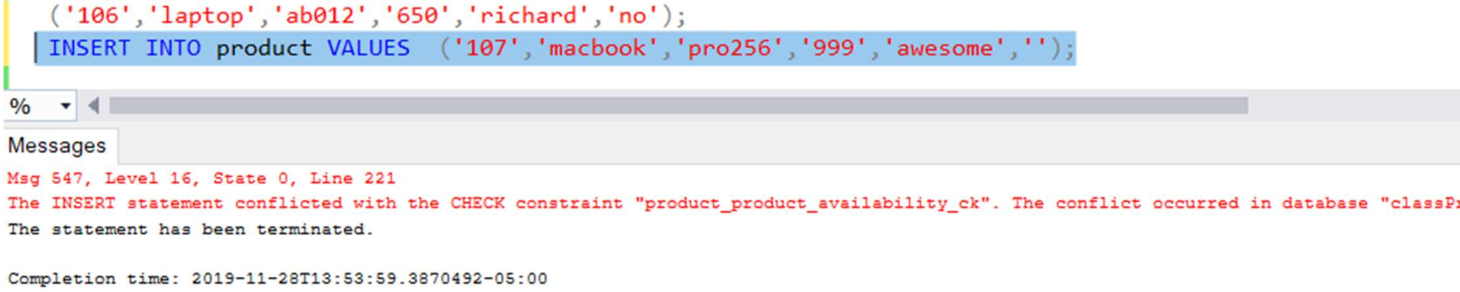
1.



```
('106','laptop','ab012','650','richard','no');
INSERT INTO product VALUES  ('107','macbook','pro256','999','awesome','');
```

```
Msg 547, Level 16, State 0, Line 221
The INSERT statement conflicted with the CHECK constraint "product_product_availability_ck". The conflict occurred in database "classPr
The statement has been terminated.

Completion time: 2019-11-28T13:53:59.3870492-05:00
```

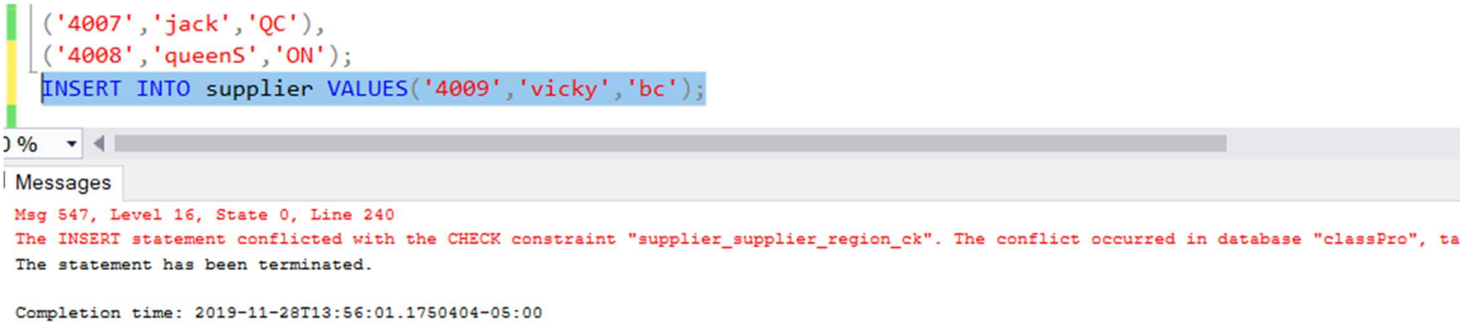
2.



```
('106','laptop','ab012','650','richard','no');
INSERT INTO product VALUES  ('107','macbook','pro256','999','awesome','');
```

```
Msg 547, Level 16, State 0, Line 221
The INSERT statement conflicted with the CHECK constraint "product_product_availability_ck". The conflict occurred in database "classPr
The statement has been terminated.

Completion time: 2019-11-28T13:53:59.3870492-05:00
```

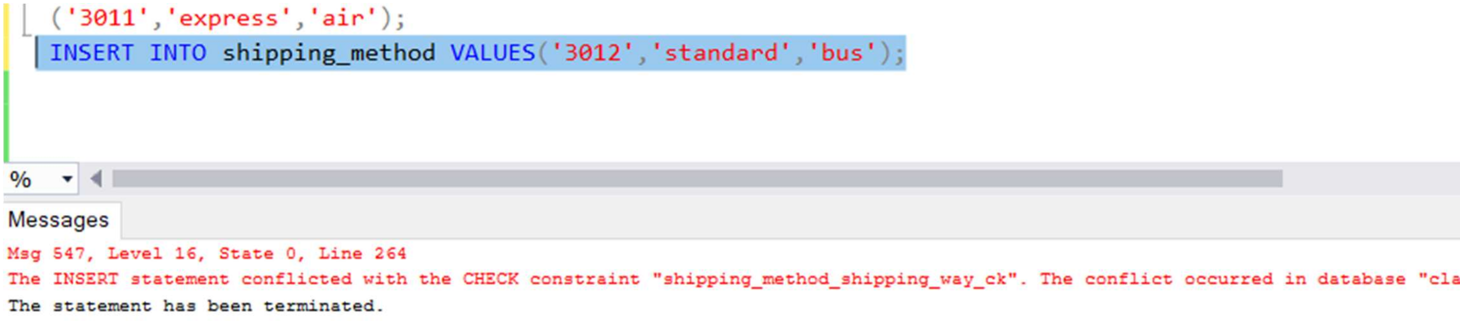
3.



```
('4007','jack','QC'),
('4008','queenS','ON');
INSERT INTO supplier VALUES('4009','vicky','bc');
```

```
Msg 547, Level 16, State 0, Line 240
The INSERT statement conflicted with the CHECK constraint "supplier_supplier_region_ck". The conflict occurred in database "classPro", ta
The statement has been terminated.

Completion time: 2019-11-28T13:56:01.1750404-05:00
```

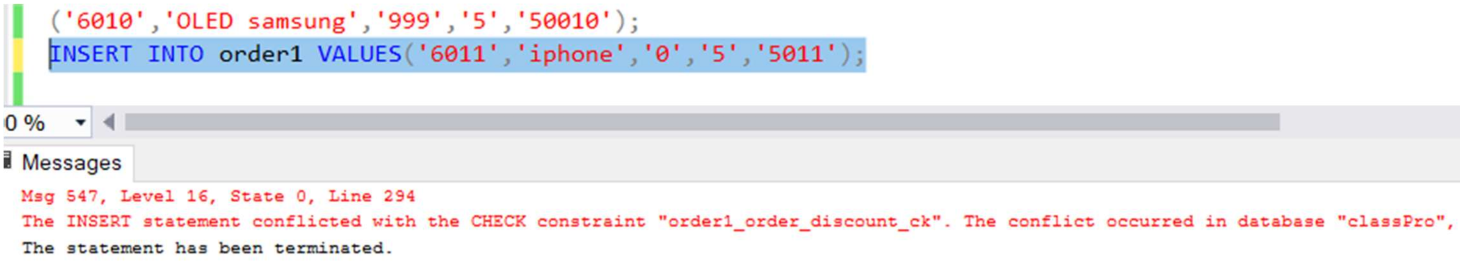
4.



```
('3011','express','air');
INSERT INTO shipping_method VALUES('3012','standard','bus');
```

```
Msg 547, Level 16, State 0, Line 264
The INSERT statement conflicted with the CHECK constraint "shipping_method_shipping_way_ck". The conflict occurred in database "cla
The statement has been terminated.
```

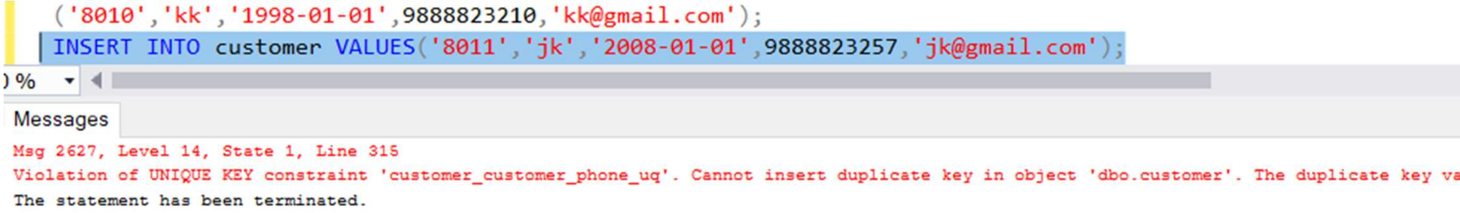
5.



```
('6010','OLED samsung','999','5','50010');
INSERT INTO order1 VALUES('6011','iphone','0','5','5011');
```

```
Msg 547, Level 16, State 0, Line 294
The INSERT statement conflicted with the CHECK constraint "order1_order_discount_ck". The conflict occurred in database "classPro",
The statement has been terminated.
```

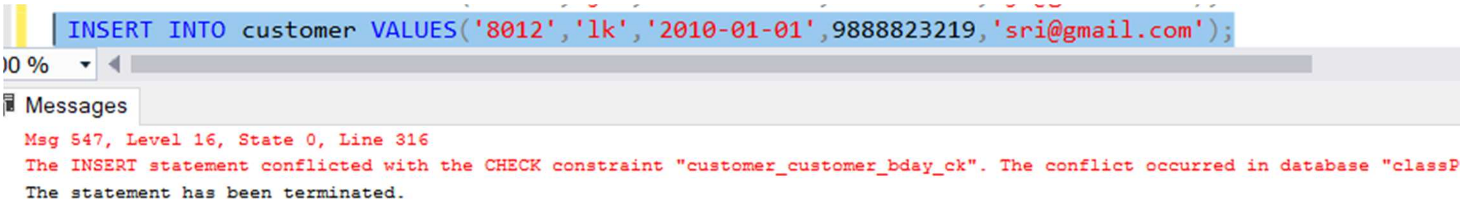
6.



```
('8010','kk','1998-01-01',9888823210,'kk@gmail.com');
INSERT INTO customer VALUES('8011','jk','2008-01-01',9888823257,'jk@gmail.com');
```

```
Msg 2627, Level 14, State 1, Line 315
Violation of UNIQUE KEY constraint 'customer_customer_phone_uq'. Cannot insert duplicate key in object 'dbo.customer'. The duplicate key va
The statement has been terminated.
```

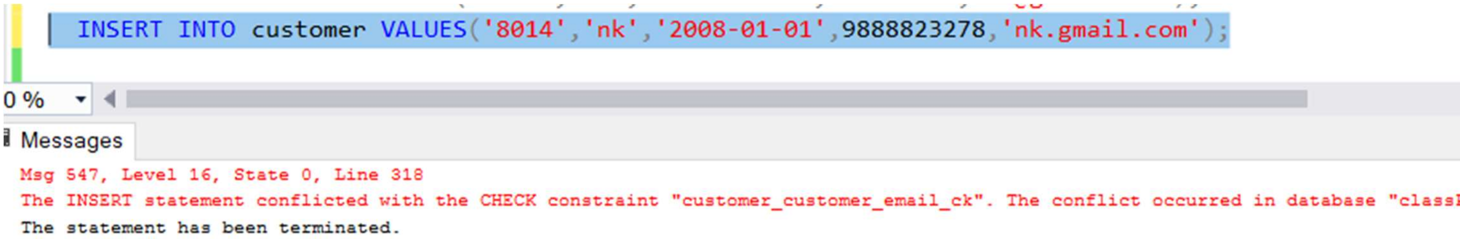
7.



```
INSERT INTO customer VALUES('8012','lk','2010-01-01',9888823219,'sri@gmail.com');
```

```
Msg 547, Level 16, State 0, Line 316
The INSERT statement conflicted with the CHECK constraint "customer_customer_bday_ck". The conflict occurred in database "classP
The statement has been terminated.
```

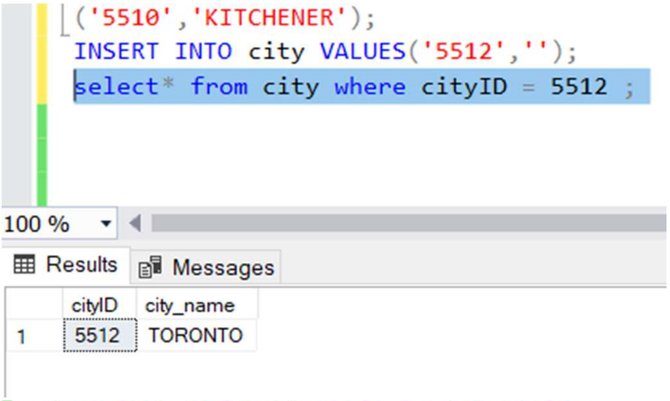
8.



```
INSERT INTO customer VALUES('8014','nk','2008-01-01',9888823278,'nk.gmail.com');
```

```
Msg 547, Level 16, State 0, Line 318
The INSERT statement conflicted with the CHECK constraint "customer_customer_email_ck". The conflict occurred in database "classI
The statement has been terminated.
```

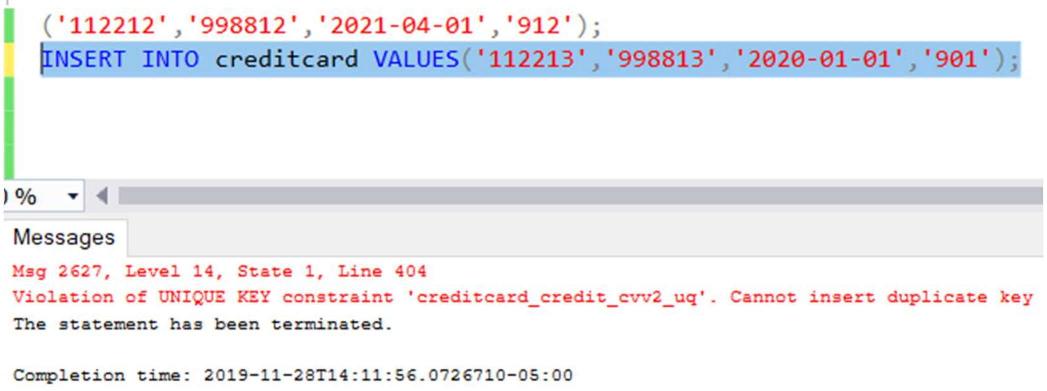
9.



```
('5510','KITCHENER');
INSERT INTO city VALUES('5512','');
select* from city where cityID = 5512 ;
```

	cityID	city_name
1	5512	TORONTO

10.



```
('112212','998812','2021-04-01','912');
INSERT INTO creditcard VALUES('112213','998813','2020-01-01','901');
```

```
Msg 2627, Level 14, State 1, Line 404
Violation of UNIQUE KEY constraint 'creditcard_credit_cvv2_uq'. Cannot insert duplicate key
The statement has been terminated.

Completion time: 2019-11-28T14:11:56.0726710-05:00
```

11.

Thank you.

References :
https://www.geeksforgeeks.org/sql-constraints/