



جامعة أم القرى
UMM AL-QURA UNIVERSITY

DATABASE **PROJECT**

2023

PHASE 1



Business rule



ER diagram



UML diagram

PROJECT NAME : SELLING PLATFORM

Project description:

The main aim of this project is to develop an easy-to-use, web-based interface for online customers that can provide them with a pleasant online shopping experience. This platform allows sellers to sell their products to the customer registered in the platform

BUSINESS RULES

The website offers several sellers to sell their products.

- Each seller must have a seller ID, a first name, a last name, an email, a phone number, an address and a password to create his selling account.

Each seller has an IBAN

The identifier for the seller is the seller ID.

A seller can sell multiple products.

The seller always has a product to sell

- Customer has multiple orders, Each order contains multiple products, one or more different products or the same product in one or more quantities

Each customer has a first and last name, one address,

The customer may have an account and not have an order

one or more phone number To communicate, and a unique ID, one email,

And a password to log in to his account.

Each customer has a credit card number ,name_on_credit card and expiration date of the card.

- Each product has a name that describes it, a special number, a specific price, a Product quantity, and information that describes it more accurately to the customer.

Each specific product has exactly one seller

- Each order has a date of order and a unique order number.

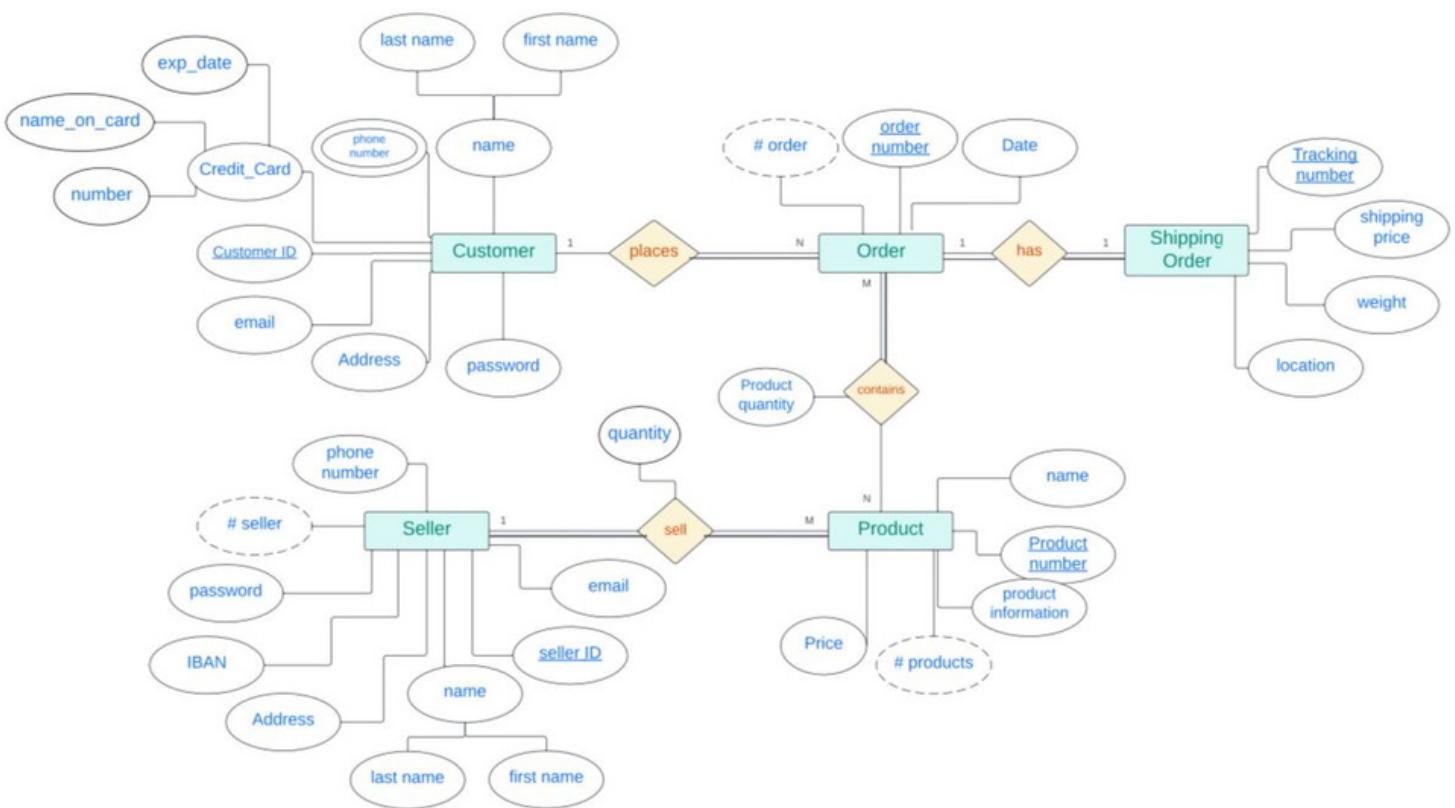
Every order will be delivered and shipped

- Each shipping order has a Tracking_number , a shipping price, weight, location.

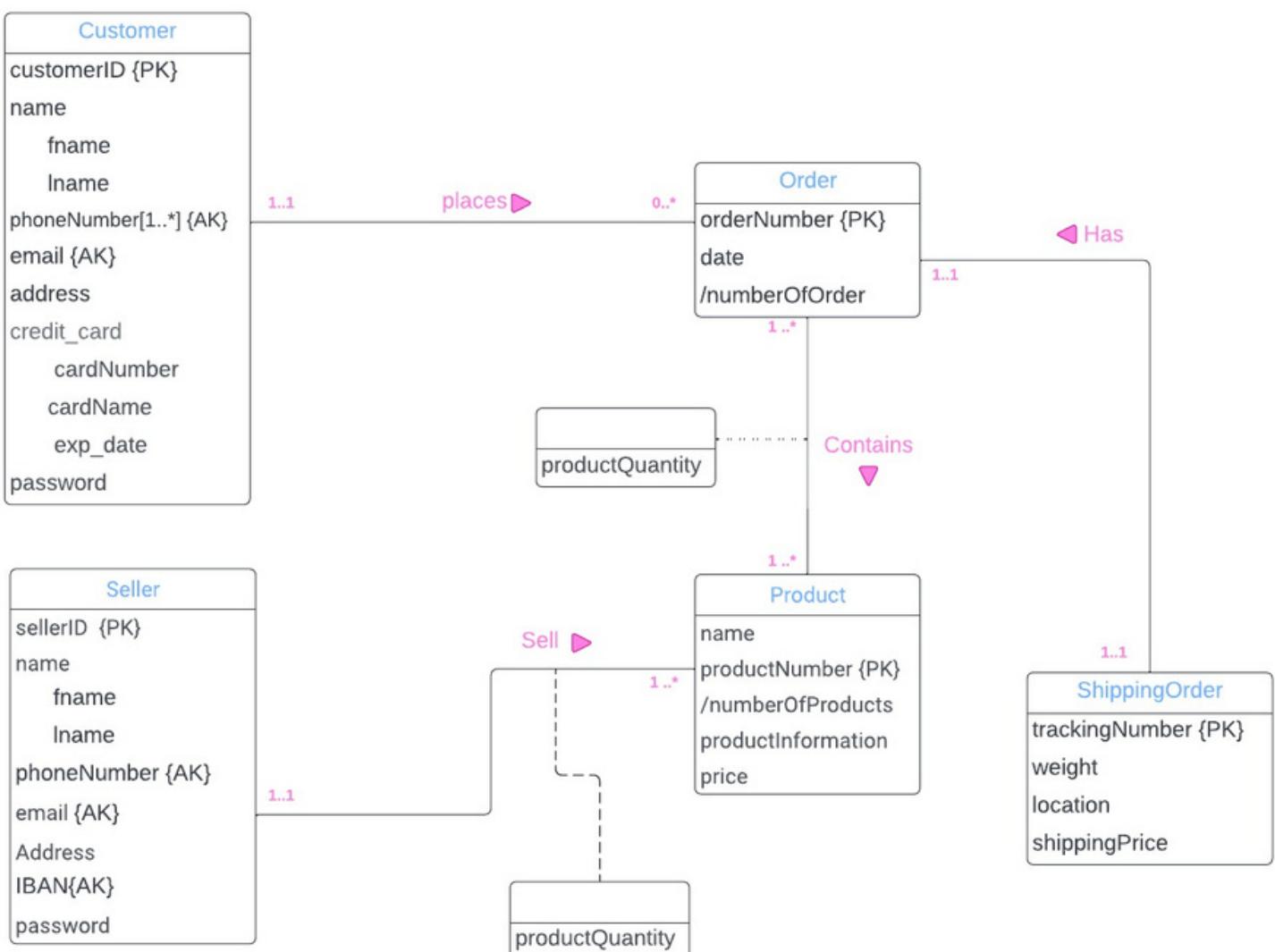
The one shipping order has one order

For every tracking number unique order number

CHEN'S NOTATION



UML DIAGRAM



PHASE 2

- Mapping of ER into Relational Schema
- Normalization

MAPPING OF ER INTO RELATIONAL SCHEMA

Step 1: Mapping of Regular Entity Types

Customer

CustomerID	firstName	LastName	email	Address	number	name_on_card	exp_date	Password
------------	-----------	----------	-------	---------	--------	--------------	----------	----------

Seller

SellerID	firstName	LastName	email	phoneNumber	IBAN	Password	Address
----------	-----------	----------	-------	-------------	------	----------	---------

Order

OrderNum	date	CustomerID	TrackingNum
----------	------	------------	-------------

Product

ProductNum	name	productInfo	Price	SellerID	quantity
------------	------	-------------	-------	----------	----------

Shipping Order

TrackingNum	shippingPrice	weight	location
-------------	---------------	--------	----------

MAPPING OF ER INTO RELATIONAL SCHEMA

Step 2: Mapping of Weak Entity Types

None

Step 3: Mapping of Binary 1:1 Relationship Types

Customer

CustomerID	firstName	LastName	email	Address	number	name_on_card	exp_date	Password
------------	-----------	----------	-------	---------	--------	--------------	----------	----------

Seller

SellerID	firstName	LastName	email	phoneNumber	IBAN	Password	Address
----------	-----------	----------	-------	-------------	------	----------	---------

Order

OrderNum	date	CustomerID	TrackingNum
----------	------	------------	-------------



Product

ProductNum	name	productInfo	Price	SellerID	quantity
------------	------	-------------	-------	----------	----------

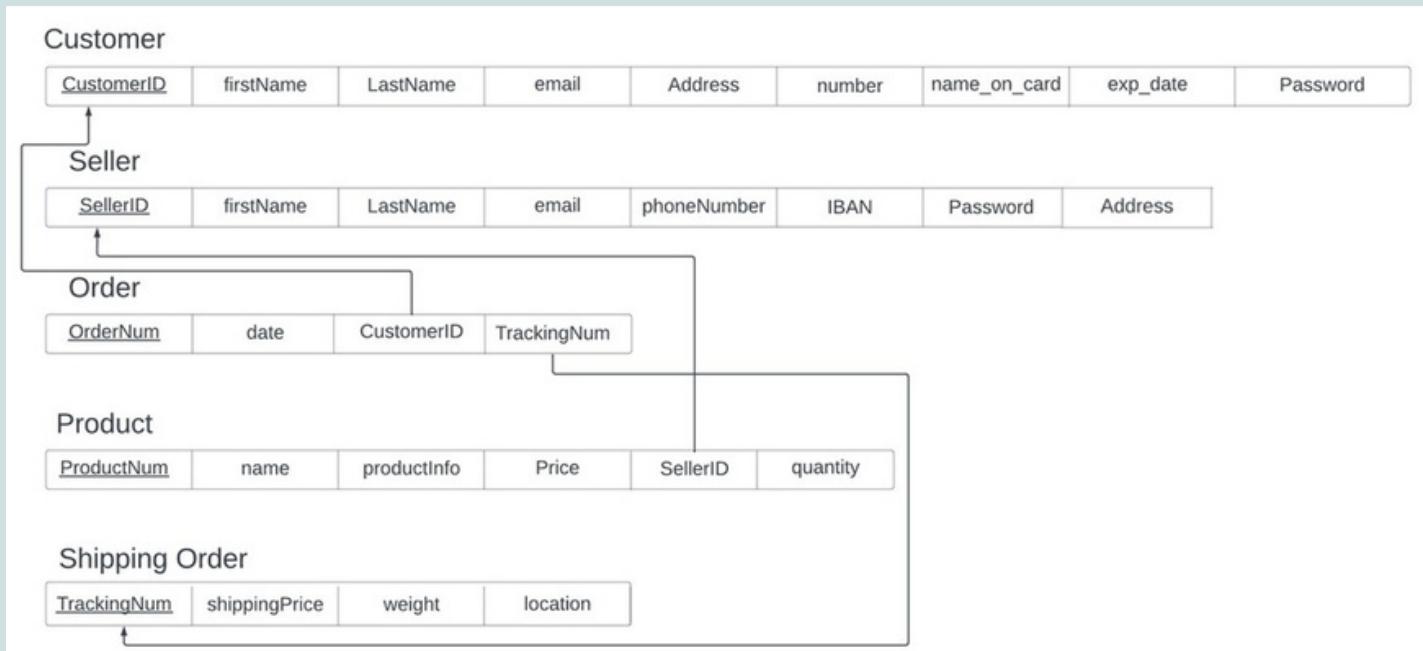
Shipping Order

TrackingNum	shippingPrice	weight	location
-------------	---------------	--------	----------



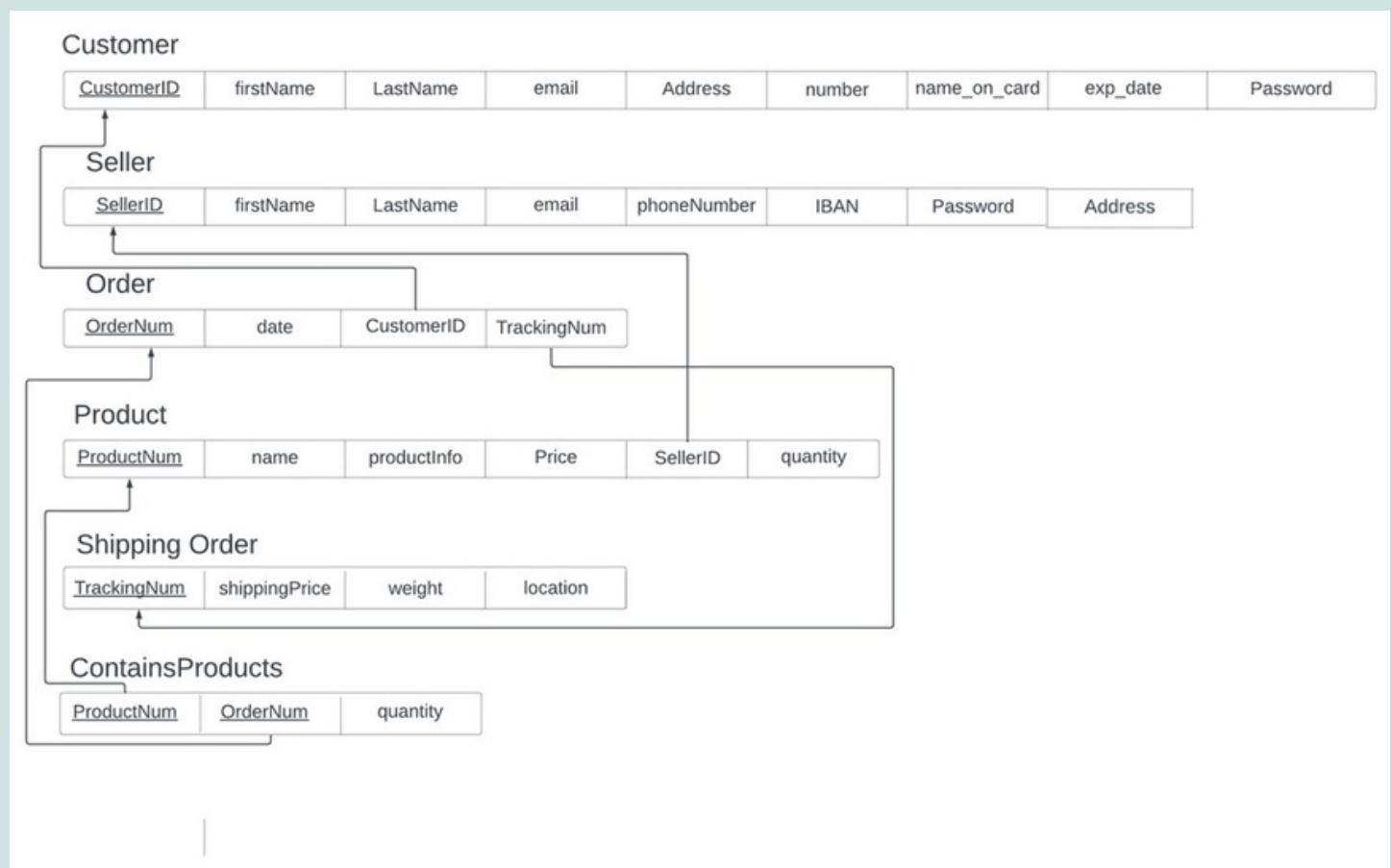
MAPPING OF ER INTO RELATIONAL SCHEMA

Step 4: Mapping of Binary 1:N Relationship Types



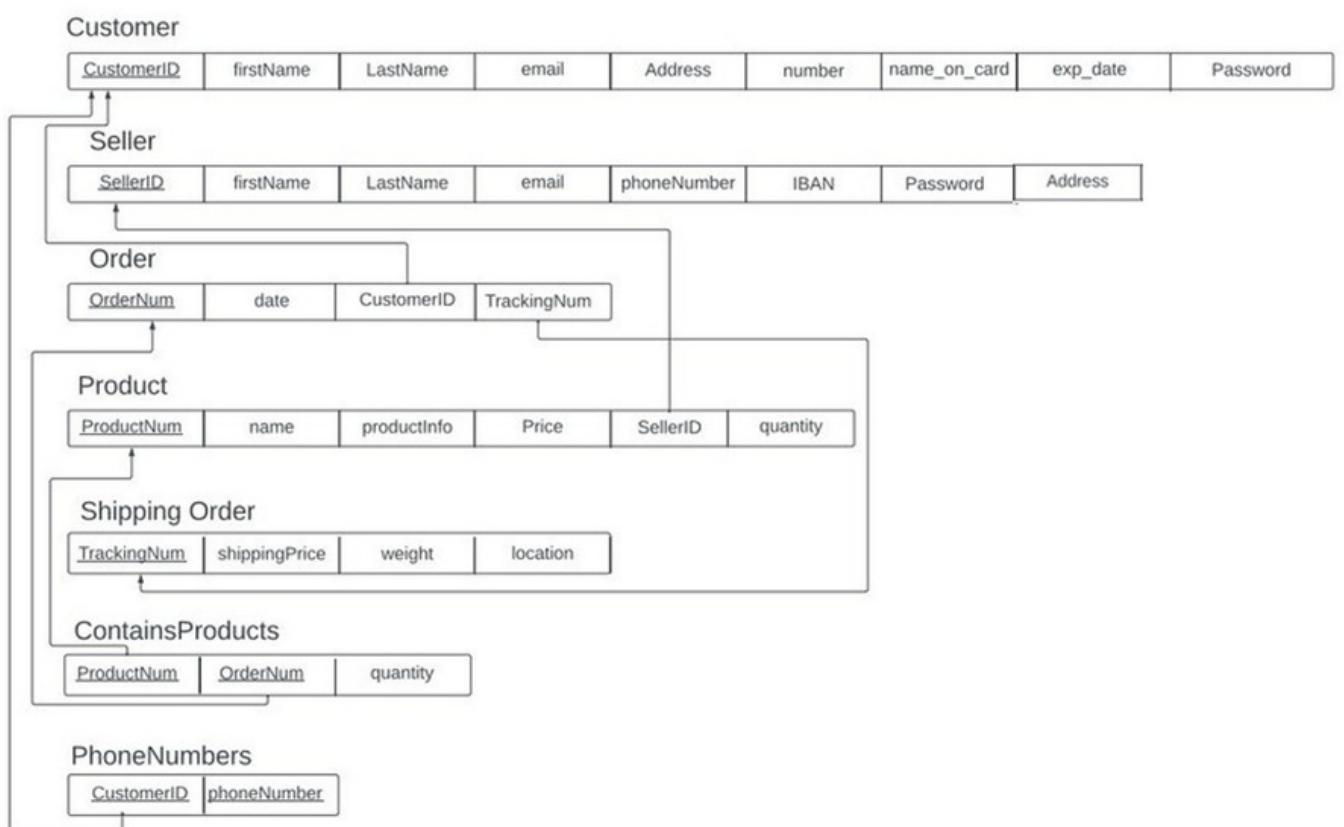
MAPPING OF ER INTO RELATIONAL SCHEMA

Step 5: Mapping of Binary M:N Relationship Types



MAPPING OF ER INTO RELATIONAL SCHEMA

Step 6: Mapping of Multivalued attributes

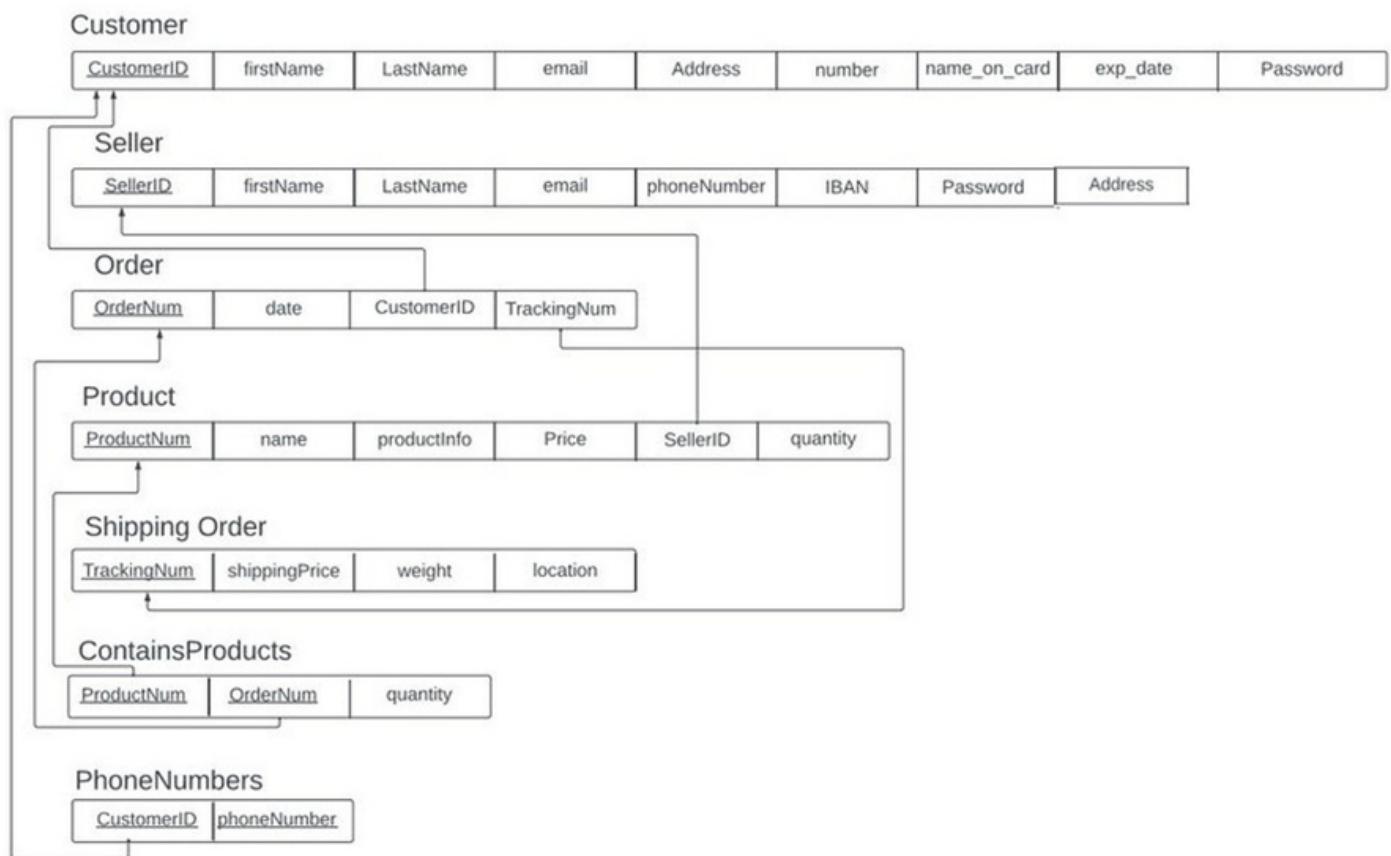


Step 7: Mapping of N-ary Relationship Types

None

MAPPING OF ER INTO RELATIONAL SCHEMA

Final mapping



NORMALIZATION

Customer

CustomerID	Fname	Lname	email	Address	CardNumber	Name-on Card	Exp-date	password
54738	Danah	Anas	Danah@gmail.com	Riyadh	1234567891234567	Danah Anas	06/23	2345@Danah
29407	Ahmed	Fahad	Ahmed@gmail.com	Najran	9876543211234567	Ahmed Fahad	04/24	Ahmed@9871
13582	Reem	Khaled	Reem@gmail.com	Hail	1234567893216549	Reem Khaled	12/23	8172Reem@
79463	Badr	Salem	Badr@gmail.com	Jazan	6789543219876543	Badr Salem	02/25	Badr80@01
65352	Sarah	Fahad	Sarah@gmail.com	Qassim	3217654896541230	Sarah Fahad	01/26	S@Fahad1234

Customer

CreditCard

CustomerID	Fname	Lname	email	Address	password	CardNumber	CardNumber	Name-on Card	Exp-date
54738	Danah	Anas	Danah@gmail.com	Riyadh	2345@Danah	1234567891234567	1234567891234567	Danah Anas	06/23
29407	Ahmed	Fahad	Ahmed@gmail.com	Najran	Ahmed@9871	9876543211234567	9876543211234567	Ahmed Fahad	04/24
13582	Reem	Khaled	Reem@gmail.com	Hail	8172Reem@	1234567893216549	1234567893216549	Reem Khaled	12/23
79463	Badr	Salem	Badr@gmail.com	Jazan	Badr80@01	6789543219876543	6789543219876543	Badr Salem	02/25
65352	Sarah	Fahad	Sarah@gmail.com	Qassim	S@Fahad1234	3217654896541230	3217654896541230	Sarah Fahad	01/26

Customer(CustomerID, Fname, Lname, email, Address, CardNumber, Name-on Card, Exp-date, password)

No multi-valued attributes or repeating groups

1NF

Customer(CustomerID, Fname, Lname, email, Address, CardNumber, Name-on Card, Exp-date, password)

No partial dependencies

2NF

There exist transitive dependencies

CreditCard(CardNumber, Name-on Card, Exp-date)

Customer(CustomerID, Fname, Lname, email, Address, password, CardNumber)

NF3

NORMALIZATION

PhoneNumbers(CustomerID, PhoneNumbers)

1NF

PhoneNumbers(CustomerID, PhoneNumbers)

No partial dependencies

2NF

PhoneNumbers(CustomerID, PhoneNumbers)

No transitive dependencies

NF3

<u>CustomerID</u>	<u>phoneNumbers</u>
54738	0533187654
54738	0587612301
29407	0511098451
29407	0598238410
29407	0544434177
13582	0568222193
79463	0591251920
79463	0501623947
65352	0519835732

NORMALIZATION

Seller

<u>SellerID</u>	Fname	Lname	email	PhoneNumber	Address	IBAN	password
11338574	Majed	Saad	Majed@gmail.com	0563452674	Mecca	1345245345637656	M_12345
66779984	Abdullah	Fahad	Abdullah@gmail.com	0564428965	Jeddah	1123454565787328	Abd@323
11225588	Ali	Naser	Ali@gmail.com	0554878429	Riyadh	1345456288789001	Ali_004
32167832	Kareem	Ali	Kareem@gmail.com	0556746890	Qassim	6473425457894509	Ka.r234
11203987	Faisal	Jaber	Faisal@gmail.com	0524314783	Jeddah	8756452130082321	Fsl@100
10144658	Ahmed	Abdullah	Ahmed@hotmail.com	0532451678	Madinah	1013435672341524	Ahm34.4

Seller(SellerID, Fname, Lname, email, PhoneNumber, Address, IBAN, password)

No multi-valued attributes or repeating groups

1NF

Seller(SellerID, Fname, Lname, email, PhoneNumber, Address, IBAN, password)

No partial dependencies

2NF

Seller(SellerID, Fname, Lname, email, PhoneNumber, Address, IBAN, password)

No partial dependencies

NF3

NORMALIZATION

Order

<u>OrderNum</u>	CustomerID	TrackingNum	date
1	54738	AY1223456789	12/3/2022
2	29407	AM6987654212	25/8/2021
3	13582	SA1234567890	8/5/2021
4	79463	HJ1974368346	3/6/2022
5	65352	BN984213567	22/9/2022

Order (OrderNum, CustomerID, TrackingNum, date)

No multi-valued attributes or repeating groups

1NF

Order (OrderNum, CustomerID, TrackingNum, date)

No partial dependencies

2NF

Order (OrderNum, CustomerID, TrackingNum, date)

No transitive dependencies

3NF

NORMALIZATION

Contains Products

<u>ProductNum</u>	<u>OrderNum</u>	quantity
8745421568	1	12
9872435745	2	20
5476577432	3	36
4688785425	4	22
8753246754	5	18

ContainsProducts (ProductNum, OrderNum, quantity)

No multi-valued attributes or repeating groups

1NF

ContainsProducts (ProductNum, OrderNum, quantity)

No partial dependencies

2NF

ContainsProducts (ProductNum, OrderNum, quantity)

No transitive dependencies

3NF

NORMALIZATION

Product

ProductNum	Name	ProductInfo	Price	SellerID	Quantity
8745421568	AIRPODS	"Color: white,"	800	11338574	12
9872435745	APPLE WATCH	"Color: Black,"	1600	66779984	20
5476577432	IPHONE 14	"Color: Purple, 256GB"	5000	11225588	36
4688785425	IPHONE 14	"Color: Purple, 128GB"	4399	11225588	22
8753246754	APPLE WATCH	"Color: Black,"	1599	11338574	18

ProductData

ProductNum	Name	ProductInfo	Price
8745421568	AIRPODS	"Color: white,"	800
9872435745	APPLE WATCH	"Color: Black,"	1600
5476577432	IPHONE 14	"Color: Purple, 256GB"	5000
4688785425	IPHONE 14	"Color: Purple, 128GB"	4399
8753246754	APPLE WATCH	"Color: Black,"	1599

Product

ProductNum	SellerID	Quantity
8745421568	11338574	12
9872435745	66779984	20
5476577432	11225588	36
4688785425	11225588	22
8753246754	11338574	18

Product (ProductNum, Name, ProductInfo, Price, SellerID, Quantity)

No multi-valued attributes or repeating groups
1NF

Product (ProductNum, Name, ProductInfo, Price, SellerID, Quantity)

there exist partial dependencies.

Product (ProductNum, SellerID, Quantity)
ProductData (ProductNum, Name, ProductInfo, Price)

2NF

Product (ProductNum, SellerID, Quantity)
ProductData (ProductNum, Name, ProductInfo, Price)

No transitive dependencies
3NF

NORMALIZATION

Shipping Order

TrackingNum	Weight	ShippingPrice	Location
AY1223456789	2Kg	20	Al-buhayrat
AM6987654212	6Kg	30	Al-nawwariyah
SA1234567890	4Kg	25	Al-zahir
HJ1974368346	2Kg	20	Al-shoqiyah
BN984213567	4Kg	25	Al-zahir

Shipping Order

TrackingNum	Weight	Location
AY1223456789	2Kg	Al-buhayrat
AM6987654212	6Kg	Al-nawwariyah
SA1234567890	4Kg	Al-zahir
HJ1974368346	2Kg	Al-shoqiyah
BN984213567	4Kg	Al-zahir

WeightPrice

Weight	ShippingPrice
2Kg	20
6Kg	30
4Kg	25
2Kg	20
4Kg	25

ShippingOrder (TrackingNum, Weight, shippingPrice, Location)

No multi-valued attributes or repeating groups

1NF

ShippingOrder (TrackingNum, Weight, shippingPrice, Location)

No partial dependencies

2NF

ShippingOrder (TrackingNum, Weight, shippingPrice, Location)

there exist transitive dependencies.

ShippingOrder (TrackingNum, Weight, Location)

WeightPrice(Weight, shippingPrice)

3NF

PHASE 3



SQL definition commands in MySQL
Workbench

SQL DEFINITION COMMANDS IN MYSQL WORKBENCH (CREATE COMMAND)

1- create schema

```
1  CREATE DATABASE IF NOT EXISTS `SellingPlatform2`;
2 • USE `SellingPlatform2`;
3
```

2- create table creditCard_Customer

```
2
3  -- creditCard_Customer TABLE
4 • CREATE TABLE IF NOT EXISTS `creditCard_Customer` (
5    `cardNaumber`          BIGINT(16)   CHECK(LENGTH(`cardNaumber`)=16),
6    `name_onCard`          VARCHAR(60)  NOT NULL,
7    `Exp_date`              DATE        CHECK(`Exp_date` > '23/6/30'),
8    CONSTRAINT `creditCard_PK` PRIMARY KEY (`cardNaumber`)
9 );
10
11
12 -- Customer TABLE
13 • CREATE TABLE IF NOT EXISTS `Customer` (
14    `customerID`           INT(10)      ,
15    `Fname`                VARCHAR(30)  NOT NULL ,
16    `Lname`                VARCHAR(30)  ,
17    `email`                VARCHAR(60)  CHECK(`email` LIKE '%gmail.com' OR `email` LIKE '%hotmail.com'),
18    `Address`              VARCHAR(50)  NOT NULL,
19    `Password`             VARCHAR(10)  CHECK(LENGTH(`Password`)>7 AND LENGTH(`Password`)<11),
20    `cardNaumber`          BIGINT(16)  NOT NULL UNIQUE ,
21
22    CONSTRAINT `Customer_PK` PRIMARY KEY (`customerID`),
23    CONSTRAINT `creditCard_FK` FOREIGN KEY (`cardNaumber`) REFERENCES `creditCard_Customer`(`cardNaumber`)
24    ON DELETE CASCADE
25    ON UPDATE CASCADE
26
27
28 );
29
30
31 • CREATE TABLE IF NOT EXISTS `PhoneNumbers` (
32    `CustomerID`           INT(5)      ,
33    `phoneNumbers`          INT(9)      CHECK(LENGTH(`phoneNumbers`)=9),
34    CONSTRAINT `PhoneNumbers_PK` PRIMARY KEY (`CustomerID`, `phoneNumbers`),
35    CONSTRAINT `PhoneNumbers_FK1` FOREIGN KEY (`CustomerID`) REFERENCES `Customer`(`CustomerID`)
36    ON DELETE CASCADE
37    ON UPDATE CASCADE
38 );
```

3-create table Customer

```
2
3  -- creditCard_Customer TABLE
4 • CREATE TABLE IF NOT EXISTS `creditCard_Customer` (
5    `cardNaumber`          BIGINT(16)   CHECK(LENGTH(`cardNaumber`)=16),
6    `name_onCard`          VARCHAR(60)  NOT NULL,
7    `Exp_date`              DATE        CHECK(`Exp_date` > '23/6/30'),
8    CONSTRAINT `creditCard_PK` PRIMARY KEY (`cardNaumber`)
9 );
10
11
12 -- Customer TABLE
13 • CREATE TABLE IF NOT EXISTS `Customer` (
14    `customerID`           INT(10)      ,
15    `Fname`                VARCHAR(30)  NOT NULL ,
16    `Lname`                VARCHAR(30)  ,
17    `email`                VARCHAR(60)  CHECK(`email` LIKE '%gmail.com' OR `email` LIKE '%hotmail.com'),
18    `Address`              VARCHAR(50)  NOT NULL,
19    `Password`             VARCHAR(10)  CHECK(LENGTH(`Password`)>7 AND LENGTH(`Password`)<11),
20    `cardNaumber`          BIGINT(16)  NOT NULL UNIQUE ,
21
22    CONSTRAINT `Customer_PK` PRIMARY KEY (`customerID`),
23    CONSTRAINT `creditCard_FK` FOREIGN KEY (`cardNaumber`) REFERENCES `creditCard_Customer`(`cardNaumber`)
24    ON DELETE CASCADE
25    ON UPDATE CASCADE
26
27
28 );
29
30
31 • CREATE TABLE IF NOT EXISTS `PhoneNumbers` (
32    `CustomerID`           INT(5)      ,
33    `phoneNumbers`          INT(9)      CHECK(LENGTH(`phoneNumbers`)=9),
34    CONSTRAINT `PhoneNumbers_PK` PRIMARY KEY (`CustomerID`, `phoneNumbers`),
35    CONSTRAINT `PhoneNumbers_FK1` FOREIGN KEY (`CustomerID`) REFERENCES `Customer`(`CustomerID`)
36    ON DELETE CASCADE
37    ON UPDATE CASCADE
38 );
```

4-create table PhoneNumbers

```
2
3  -- creditCard_Customer TABLE
4 • CREATE TABLE IF NOT EXISTS `creditCard_Customer` (
5    `cardNaumber`          BIGINT(16)   CHECK(LENGTH(`cardNaumber`)=16),
6    `name_onCard`          VARCHAR(60)  NOT NULL,
7    `Exp_date`              DATE        CHECK(`Exp_date` > '23/6/30'),
8    CONSTRAINT `creditCard_PK` PRIMARY KEY (`cardNaumber`)
9 );
10
11
12 -- Customer TABLE
13 • CREATE TABLE IF NOT EXISTS `Customer` (
14    `customerID`           INT(10)      ,
15    `Fname`                VARCHAR(30)  NOT NULL ,
16    `Lname`                VARCHAR(30)  ,
17    `email`                VARCHAR(60)  CHECK(`email` LIKE '%gmail.com' OR `email` LIKE '%hotmail.com'),
18    `Address`              VARCHAR(50)  NOT NULL,
19    `Password`             VARCHAR(10)  CHECK(LENGTH(`Password`)>7 AND LENGTH(`Password`)<11),
20    `cardNaumber`          BIGINT(16)  NOT NULL UNIQUE ,
21
22    CONSTRAINT `Customer_PK` PRIMARY KEY (`customerID`),
23    CONSTRAINT `creditCard_FK` FOREIGN KEY (`cardNaumber`) REFERENCES `creditCard_Customer`(`cardNaumber`)
24    ON DELETE CASCADE
25    ON UPDATE CASCADE
26
27
28 );
29
30
31 • CREATE TABLE IF NOT EXISTS `PhoneNumbers` (
32    `CustomerID`           INT(5)      ,
33    `phoneNumbers`          INT(9)      CHECK(LENGTH(`phoneNumbers`)=9),
34    CONSTRAINT `PhoneNumbers_PK` PRIMARY KEY (`CustomerID`, `phoneNumbers`),
35    CONSTRAINT `PhoneNumbers_FK1` FOREIGN KEY (`CustomerID`) REFERENCES `Customer`(`CustomerID`)
36    ON DELETE CASCADE
37    ON UPDATE CASCADE
38 );
```

SQL DEFINITION COMMANDS IN MYSQL WORKBENCH (CREATE COMMAND)

5- create table
CreditCard_Seller

```
140 • CREATE TABLE `CreditCard_Seller` (
141     `CardNumber` BIGINT(16) NOT NULL UNIQUE CHECK(LENGTH(`CardNumber`) = 16),
142     `Name_onCard` VARCHAR(50) NOT NULL,
143     `Exp_date` date NOT NULL,
144
145     CONSTRAINT `CreditCard_Seller_PK` PRIMARY KEY (`CardNumber`)
146 );
147
148 • CREATE TABLE `Seller` (
149     `sellerID` BIGINT(8) NOT NULL UNIQUE,
150     `Fname` VARCHAR(50) NOT NULL,
151     `Lname` VARCHAR(50),
152     `email` VARCHAR(50) NOT NULL UNIQUE CHECK(`email` LIKE ('%@gmail.com') OR `email` LIKE ('%@hotmail.com') OR `email` LIKE ('%@ymail.com')),
153     `PhoneNumber` VARCHAR(10) NOT NULL CHECK(LENGTH(`PhoneNumber`) = 10),
154     `Address` VARCHAR(50) NOT NULL,
155     `password` VARCHAR(8) NOT NULL CHECK(LENGTH(`password`) = 7),
156
157     `CardNumber` BIGINT(16) NOT NULL UNIQUE ,
158
159     CONSTRAINT `Seller_PK` PRIMARY KEY (`sellerID`),
160     CONSTRAINT `Seller_FK1` FOREIGN KEY (`CardNumber`) REFERENCES `CreditCard_Seller`(`CardNumber`)
161     ON DELETE CASCADE
162     ON UPDATE CASCADE
163 );
```

6- create table Seller

7- create table
ProductData

```
244 • CREATE TABLE `ProductData`(
245     `ProductNum` BIGINT(12) NOT NULL,
246     `Name` VARCHAR(50) NOT NULL,
247     `ProductInfo` VARCHAR(500),
248     `Price` DECIMAL(10,3) NOT NULL,
249     CONSTRAINT `ProductData_PK` PRIMARY KEY(ProductNum)
250 );
251
252 • CREATE TABLE `Product`(
253     `ProductNum` BIGINT(12),
254     `sellerID` BIGINT(8) NOT NULL,
255     `Quantity` INT(3) NOT NULL,
256     CONSTRAINT `Product_PK` PRIMARY KEY(`ProductNum`,`SellerId`),
257     CONSTRAINT `Product_FK` FOREIGN KEY(`ProductNum`) -- FK1
258     REFERENCES `ProductData`(`ProductNum`),
259     CONSTRAINT `Product_FK2` FOREIGN KEY(`sellerID`) -- FK2
260     REFERENCES `Seller`(`sellerID`)
261     ON DELETE CASCADE
262     ON UPDATE CASCADE
263 );
```

8-create table Product

SQL DEFINITION COMMANDS IN MYSQL WORKBENCH (CREATE COMMAND)

9- create table weightPrice

```
301 • Ⓜ create table `weightPrice`(
302     `weight`      INT(3) NOT NULL,
303     `shippingPrice`  INT(3) NOT NULL,
304
305     CONSTRAINT `weightPrice_PK` PRIMARY KEY (`weight`)
306 );
307
308 • Ⓜ create table `shippingOrder`(
309     TrackingNum VARCHAR(12) NOT NULL,
310     weight INT(3),
311     location VARCHAR(30) NOT NULL,
312
313     CONSTRAINT `shippingOrder_PK` PRIMARY KEY (TrackingNum,`weight`),
314
315     CONSTRAINT `weight_FK2` FOREIGN KEY (`weight`)
316         REFERENCES `weightPrice` (`weight`)
317         ON DELETE CASCADE
318         ON UPDATE CASCADE
319 );
```

10- create table shippingOrder

11- create table Order

```
366    -- ORDER TABLE
367 • Ⓜ CREATE TABLE `Order`(
368     `OrederNum` INT(10) NOT NULL,
369     `date` date ,
370     `CustomerID` INT(10) NOT NULL,
371     `TrackingNum` VARCHAR(11) NOT NULL unique,
372     constraint PK PRIMARY KEY (OrederNum),
373     constraint FK2 FOREIGN KEY (customerID) references Customer(customerID), -- FK --
374     constraint FK3 FOREIGN KEY (TrackingNum) references shippingOrder(TrackingNum) -- FK --
375     ON DELETE CASCADE
376     ON UPDATE CASCADE );
```

12- create table ContainsProducts

```
378    -- Contains Products TABLE
379 • Ⓜ CREATE TABLE `ContainsProducts`(
380     `ProductNum` BIGINT(12) NOT NULL,
381     `OrederNum` INT(10) NOT NULL,
382     `quantity` INT(2),
383     constraint PK PRIMARY KEY (OrederNum,ProductNum),
384     constraint FK1 FOREIGN KEY (OrederNum) references `Order`(OrederNum), -- FK1
385     constraint FKP2 FOREIGN KEY (ProductNum) references Product(ProductNum) -- FK2
386     ON DELETE CASCADE
387     ON UPDATE CASCADE
388 );
```

SQL DEFINITION COMMANDS IN MYSQL WORKBENCH (INSERT COMMAND)

13- Insert Into creditCard_Customer

```
75 -- INSERT MULTIPLE ROWS
76 • INSERT INTO `creditCard_Customer`
77   VALUES
78   ('1234567891234567', 'Danah Anas', '23/7/1'),
79   ('9876543211234567', 'Ahmed Fahad ', '24/4/1'),
80   ('1234567893216549', 'Reem Khaled', '23/12/1'),
81   ('6789543219876543', 'Badr Salem', '25/2/1'),
82   ('3217654896541230', 'Sarah Fahad', '26/1/1');
83 • INSERT INTO `Customer`
84   VALUES
85   ('54738', 'Danah', 'Anas', 'Danah@gmail.com', 'Riyadh', '2345@Danah', '1234567891234567'),
86   ('29407', 'Ahmed', 'Fahad', 'Ahmed@gmail.com', 'Najran', 'Ahmed@9871', '9876543211234567'),
87   ('13582', 'Reem', 'Khaled', 'Reem@gmail.com', 'Hail', '8172Reem@', '1234567893216549'),
88   ('79463', 'Badr', 'Salem', 'Badr@gmail.com', 'Jazan', 'Badr0@01', '6789543219876543'),
89   ('65352', 'Sarah', 'Fahad', 'Sarah@gmail.com', 'Qassim', 'S@Fahad123', '3217654896541230');
```

14-Insert Into Customer

```
92 • INSERT INTO `creditCard_Customer`
93   VALUES
94   ('1239382391045670', 'lami Ahmed', '28/2/6'),
95   ('1297382391045670', 'daleen amjad', '24/1/9');
96 • INSERT INTO `Customer`
97   VALUES
98   ('34983', 'lami', 'Ahmed', 'lami@hotmail.com', 'Riyadh', '349@lami', '1239382391045670'),
99   ('20915', 'daleen', 'amjad', 'daleen@hotmail.com', 'Qassim', '51@daleen', '1297382391045670');
```

15- Insert date into PhoneNumbers

```
57
58
59 • INSERT INTO `phonenumbers`
60   VALUES
61   ('54738 ', '0533187654'),
62   ('54738 ', '0587612301'),
63   (' 29407 ', '0511098451'),
64   (' 29407 ', '0598238410'),
65   (' 29407 ', '0544434177'),
66   (' 13582', '0568222193'),
67   ('79463 ', '0591251920'),
68   ('79463 ', '0501623947'),
69   ('65352', '0519835732'),
70   ('34983', '0533498354'),
71   ('20915', '0502915301');
```

16-insert date into creditCardSeller

```
165 • INSERT INTO `CreditCard_Seller`(`CardNumber`, `Name_onCard`, `Exp_date`)
166   VALUES(1345245345637656, 'Majed Saad', STR_TO_DATE('2023-08-01', '%Y-%m-%d')),
167   (1123454565787328, 'Abdullah Fahad', STR_TO_DATE('2024-01-01', '%Y-%m-%d')),
168   (1345456288789001, 'Ali Naser', STR_TO_DATE('2023-10-01', '%Y-%m-%d')),
169   (6473425457894509, 'Kareem Ali', STR_TO_DATE('2023-03-01', '%Y-%m-%d')),
170   (8756452130082321, 'Faisal Jaber', STR_TO_DATE('2024-02-01', '%Y-%m-%d')),
171   (1013435672341524, 'Ahmed Abdullah', STR_TO_DATE('2023-6-01', '%Y-%m-%d'));
172
173
174 • INSERT INTO `Seller`(`sellerID`, `Fname`, `Lname`, `email`, `PhoneNumber`, `Address`, `password`, `CardNumber`)
175   VALUES(1138574, 'Majed', 'Saad', 'Majed@gmail.com', '0563452674', 'Mecca', 'MA_1234', 1345245345637656),
176   (66779984, 'Abdullah', 'Fahad', 'Abdullah@gmail.com', '0564428965', 'Jeddah', 'Abdg323', 1123454565787328),
177   (11225588, 'Ali', 'Naser', 'Ali@hotmail.com', '0554878429', 'Riyadh', 'Ali_004', 1345456288789001),
178   (32167832, 'Kareem', 'Ali', 'Kareem@gmail.com', '0556746890', 'Qassim', 'Ka_r234', 6473425457894509),
179   (11203987, 'Faisal', 'Jaber', 'Faisal@gmail.com', '0524314783', 'Jeddah', 'Fs10100', 8756452130082321),
180   (10144658, 'Ahmed', 'Abdullah', 'Ahmed@hotmail.com', '0532451678', 'Madinah', 'Ahm34.4', 1013435672341524);
```

17-Insert data into Seller

SQL DEFINITION COMMANDS IN MYSQL WORKBENCH (INSERT COMMAND)

18-Insert data into ProductData , Prodcut

```
265 • INSERT INTO `ProductData` (`ProductNum`, `Name`, `ProductInfo`, `Price`)
266   VALUES(8346512303,'Samsung Galaxy Buds Pro','Samsung Galaxy Buds Pro,Phantom Violet',577),
267   (3547482928,'Samsung Galaxy Buds Plus','Samsung Galaxy Buds Plus, White',549),
268   (2345565657,'Samsung Galaxy Z Flip 4','Samsung Galaxy Z Flip 4, 5G, 256GB, Bora Purple',3999),
269   (3452633736,'Samsung Galaxy Z Flip 4','Samsung Galaxy Z Flip 4, 5G, 256GB, Blue',3999),
270   (1325389630,'Samsung Galaxy Watch 5','Samsung Galaxy Watch 5, 40MM, Silver',1099);
271
272 • INSERT INTO `ProductData` (`ProductNum`, `Name`, `ProductInfo`, `Price`)
273   VALUES (8745421568,'Apple AirPods','Color:white,2nd Gen with Charging case',800),
274   (9872435745,'Apple Watch','Series 8 GPS 45mm RED Case With RED Sport Band',1649),
275   (5476577432,'iPhone 14','iPhone 14 Pro Max ,5G,256GB,Deep Purple',5000),
276   (4688785425,'iPhone 14','iPhone 14 Pro Max ,5G,128GB,Deep Purple',4499),
277   (8753246754,'Apple Watch','Series 8 GPS 41mm Midnight Case With Midnight Sport Band',1549);
278
279 • INSERT INTO `Product` (`ProductNum`, `SellerId`, `Quantity`)
280   VALUES (8346512303,32167832,14),(3547482928,11203987,20),(2345565657,10144658,18)
281   ,(3452633736,10144658,42),(1325389630,10144658,19);
282
283 • INSERT INTO `Product` (`ProductNum`, `SellerID`, `Quantity`)
284   VALUES (8745421568,11338574,12),(9872435745,66779984,20),(5476577432,11225588,36)
285   ,(4688785425,11225588,22),(8753246754,11338574,18);
286 • SELECT * FROM `Product`;
```

19- Insert data into wightPrice

```
321 • INSERT INTO `weightPrice` (`weight`, `shippingPrice`) VALUES
322   (1,15),
323   (2,20),
324   (6,40),
325   (4,30),
326   (3,25),
327   (5,35),
328   (7,45),
329   (8,50),
330   (9,55),
331   (10,70);
```

20- Insert data into ShippingOrder

```
335   INSERT INTO `shippingOrder` (`TrackingNum`, `weight`, `location`) VALUES
336     ('AY1223456789',2,'Al-buhayrat'),
337     ('AM6987654212',6,'Al-nawwariyah'),
338     ('SA1234567890',4,'Al-zahir'),
339     ('HJ1974368346',3,'Al-shoqiyah'),
340     ('BN9842135678',6,'Al-zahir'),
341     ('KW1234457890',7,'Al-buhayrat');
```

21- insert into Order and ContainsProducts

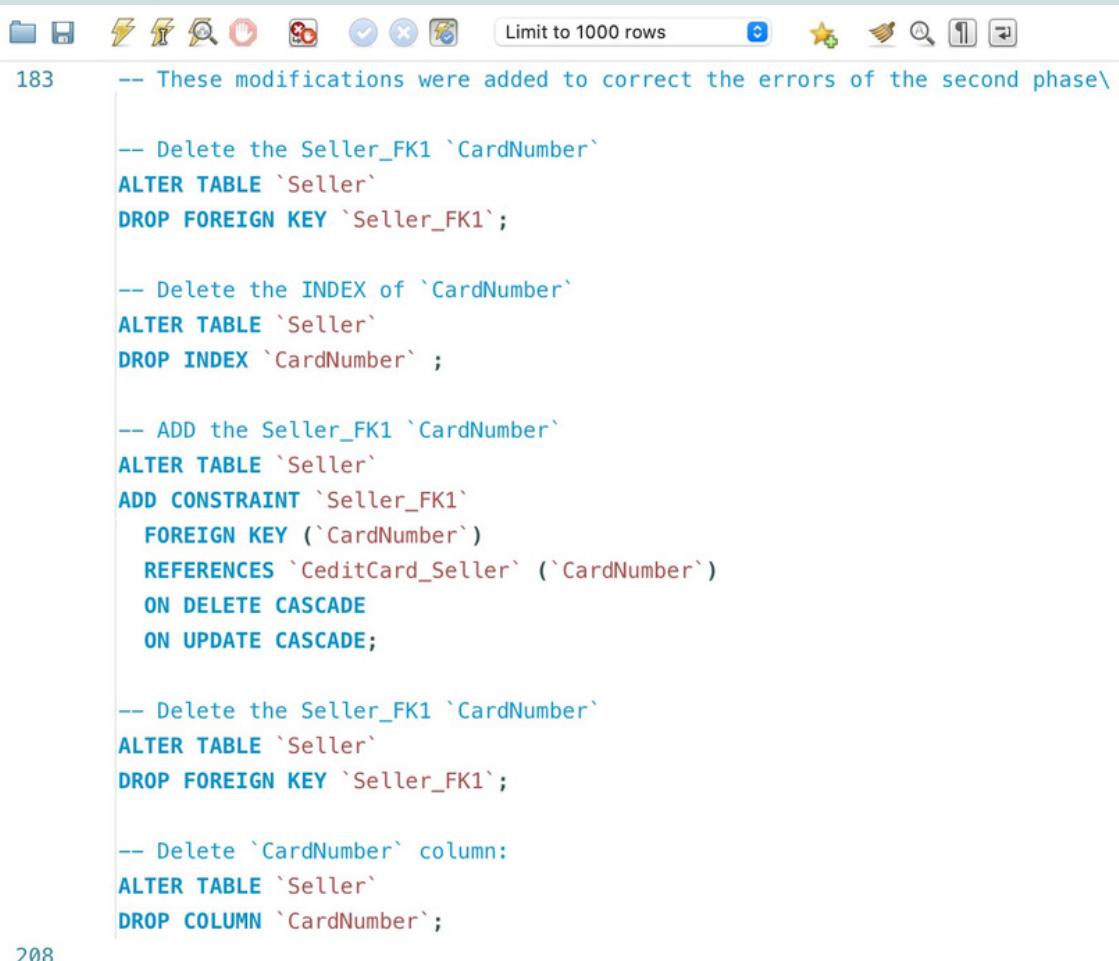
```
400 -- INSERT new row (ONE ROW) -----
401 • INSERT INTO `Order` (OrederNum,date,CustomerID,TrackingNum)
402   values(1,STR_TO_DATE('2022-03-12','%Y-%m-%d'),54730,'AY1223456789');
403 • INSERT INTO `ContainsProducts` (ProductNum,OrederNum,quantity)
404   values(8346512303,1,12);
405 • INSERT INTO `Order` (OrederNum,date,CustomerID,TrackingNum)
406   values(6,STR_TO_DATE('2023-04-04','%Y-%m-%d'),54730,'KH1234457890');
407 • INSERT INTO `ContainsProducts` (ProductNum,OrederNum,quantity)
408   values(4688785425,6,5);
409 • INSERT INTO `ContainsProducts` (ProductNum,OrederNum,quantity)
410   values(8346512303,6,9);
411 • INSERT INTO `ContainsProducts` (ProductNum,OrederNum,quantity)
412   values(5476577432,6,3);
413 -- INSERT MULTIPLE ROWS -----
414 • INSERT INTO `Order` (OrederNum,CustomerID,date,TrackingNum)
415   values(2,29407,STR_TO_DATE('2021-08-25','%Y-%m-%d'),'AH6987654212'),
416   (3,13582,STR_TO_DATE('2021-05-08','%Y-%m-%d'),'SA1234567890'),
417   (4,79463,STR_TO_DATE('2022-06-03','%Y-%m-%d'),'HJ1974368346'),
418   (5,65352,STR_TO_DATE('2022-09-22','%Y-%m-%d'),'BN9842135678');
419 • INSERT INTO `ContainsProducts` (ProductNum,OrederNum,quantity)
420   values(9872435745,2,20),(5476577432,3,36),(4688785425,4,22),(8753246754,5,18);
421 • INSERT INTO `ContainsProducts` (ProductNum,OrederNum,quantity)
422   values(8753246754,2,1),(4688785425,3,4),(9872435745,4,5),(5476577432,5,6);
```

SQL DEFINITION COMMANDS IN MYSQL WORKBENCH (DELETE ,UPDATE AND ALTER COMMANDS)

22- alter table Customer and PhoneNumbers

```
-- These modifications were added to correct the errors of the second phase
103 • ALTER TABLE `Customer` DROP CONSTRAINT `creditCard_FK`;
104 • ALTER TABLE `PhoneNumbers` DROP CONSTRAINT `PhoneNumbers_FK`;
105 • ALTER TABLE `PhoneNumbers`
106 ADD CONSTRAINT `PhoneNumbers_FK1` FOREIGN KEY (`CustomerID`) REFERENCES `Customer`(`CustomerID`)
107 ON DELETE CASCADE
108 ON UPDATE CASCADE;
109 • DROP TABLE `CustomerID`;
110
111
112
```

23 - delete and alter table Seller and then CerditCard_Seller



```
183 -- These modifications were added to correct the errors of the second phase\

-- Delete the Seller_FK1 `CardNumber`
ALTER TABLE `Seller`
DROP FOREIGN KEY `Seller_FK1`;

-- Delete the INDEX of `CardNumber`
ALTER TABLE `Seller`
DROP INDEX `CardNumber` ;

-- ADD the Seller_FK1 `CardNumber`
ALTER TABLE `Seller`
ADD CONSTRAINT `Seller_FK1`
FOREIGN KEY (`CardNumber`)
REFERENCES `CeditCard_Seller`(`CardNumber`)
ON DELETE CASCADE
ON UPDATE CASCADE;

-- Delete the Seller_FK1 `CardNumber`
ALTER TABLE `Seller`
DROP FOREIGN KEY `Seller_FK1`;

-- Delete `CardNumber` column:
ALTER TABLE `Seller`
DROP COLUMN `CardNumber`;
```

208

SQL DEFINITION COMMANDS IN MYSQL WORKBENCH (DELETE ,UPDATE AND ALTER COMMANDS)

24 - update and alter table Seller and delete CerditCard_Seller

```
208
209      -- Add `IBAN` as a new column
210 •  ALTER TABLE `Seller`
211     ADD `IBAN` BIGINT(16) NOT NULL;
212
213     -- Update the default value(0)
214 •  UPDATE `Seller`
215     SET `IBAN` = 1345245345637656
216     WHERE `sellerID` = 11338574;
217 •  UPDATE `Seller`
218     SET `IBAN` = 1123454565787328
219     WHERE `sellerID` = 66779984;
220 •  UPDATE `Seller`
221     SET `IBAN` = 1345456288789001
222     WHERE `sellerID` = 11225588;
223 •  UPDATE `Seller`
224     SET `IBAN` = 6473425457894509
225     WHERE `sellerID` = 32167832;
226 •  UPDATE `Seller`
227     SET `IBAN` = 8756452130082321
228     WHERE `sellerID` = 11203987;
229 •  UPDATE `Seller`
230     SET `IBAN` = 1013435672341524
231     WHERE `sellerID` = 10144658;
232
233     -- Delete `CeditCard_Seller` Table:
234 •  DROP TABLE `CeditCard_Seller`;
235
```

25 - delete and update table weightPrice

```
317     -- UPDATE weightPrice TABLE
318 •  UPDATE weightPrice
319     SET shippingPrice = 60
320     WHERE weight = 10;
321
322     -- DELETE A ROW FROM weightPrice TABLE WHERE WEIGHT EQUALS 1
323 •  DELETE FROM weightPrice
324     WHERE weight = 1;
```

26 - alter table Order

```
394     -- ALTER A Secheme -----
395 •  ALTER TABLE `Order`
396     modify `TrackingNum` VARCHAR(12) NOT NULL;
397 •  ALTER TABLE `ContainsProducts`
398     modify `ProductNum` BIGINT(12) NOT NULL;
399
```

SQL DEFINITION COMMANDS IN MYSQL WORKBENCH (JOIN, SUBQUERY, HAVING, WHERE ORDER BY AND GROUP BY COMMANDS)

```
289 -- Display the (Maximum Quantity For One Product) and the (Total Products Available) for each seller,
290 -- and Some seller information.
291 • SELECT `product`.SellerId, CONCAT(`Fname`, ' ', `Lname`) AS SellerFullName,
292     MAX(`Quantity`) AS MaximumQuantityForOneProduct, SUM(`Quantity`) AS TotalProductsAvailable
293 FROM `SellingPlatform2`.`Seller` -- left table
294 INNER JOIN `SellingPlatform2`.`product` -- right table
295 ON (`Seller`.SellerID = `product`.SellerId)
296 GROUP BY `product`.SellerId
297 ORDER BY MaximumQuantityForOneProduct DESC;
298
299
```

SellerId	SellerFullName	MaximumQuantityForOneProduct	TotalProductsAvailable
► 10144658	Ahmed Abdullah	42	79
11225588	Ali Naser	36	58
11203987	Faisal Jaber	20	20
66779984	Abdullah Fahad	20	20
11338574	Majed Saad	18	30
32167832	Kareem Ali	14	14

```
112
113 -- COUNT the number of customer from each city
114 • SELECT `Address`,
115 count(`Address`) AS TOTAL
116 FROM `Customer`
117 GROUP BY `Address`;
118
```

Address	TOTAL
► Hail	1
Qassim	2
Najran	1
Riyadh	2
Jazan	1

```
119 -- Displaying users who entered more than one mobile number when they registered
120 -- with a display of the total number of mobile numbers they entered
121 • SELECT `CustomerID`,
122 count(*) AS TOTAL
123 FROM `PhoneNumbers`
124 GROUP BY `CustomerID`
125 HAVING count(*) > 1;
```

CustomerID	TOTAL
► 29407	3
54738	2
79463	2

SQL DEFINITION COMMANDS IN MYSQL WORKBENCH (JOIN, SUBQUERY, HAVING, WHERE ORDER BY AND GROUP BY COMMANDS)

```
127 -- VIEW USERS WHO USE GMAIL
128 • SELECT `Fname` , `CustomerID` , `email`
129   FROM `Customer`
130   WHERE (`email` LIKE '%gmail.com');
131
```

	Fname	CustomerID	email
▶	Reem	13582	Reem@gmail.com
◀	Ahmed	29407	Ahmed@gmail.com
▶	Danah	54738	Danah@gmail.com
◀	Sarah	65352	Sarah@gmail.com
▶	Badr	79463	Badr@gmail.com

```
-- 
132 -- VIEW USERS WHO USE hotmail
133 • SELECT `Fname` , `CustomerID` , `email`
134   FROM `Customer`
135   WHERE (`email` LIKE '%hotmail.com');
```

100% 1:457

Result Grid Filter Rows: Search

	Fname	CustomerID	email
▶	daleen	20915	daleen@hotmail.com
◀	lami	34983	lami@hotmail.com

```
327 -- DISPLAY weightPrice TABLE IN ASCENDING ORDER
328 • SELECT *
329   FROM weightPrice
330   ORDER BY weight ASC;
331
```

	weight	shippingPrice
▶	2	20
◀	3	25
▶	4	30
◀	5	35
▶	6	40
◀	7	45
▶	8	50
◀	9	55
▶	10	60

SQL DEFINITION COMMANDS IN MYSQL WORKBENCH [JOIN, SUBQUERY, HAVING, WHERE AND GROUP BY COMMANDS]

```
332    -- DISPLAY shippingOrder TABLE IN DESCENDING ORDER
333 •  SELECT *
334    FROM shippingOrder
335    ORDER BY weight DESC;
336
```

	TrackingNum	weight	location
▶	KW1234457890	7	Al-buhayrat
◀	AM6987654212	6	Al-nawwariyah
◀	BN9842135678	6	Al-zahir
◀	SA1234567890	4	Al-zahir
◀	HJ1974368346	3	Al-shoqiyah
◀	AY1223456789	2	Al-buhayrat

```
338    -- DISPLAY HOW MANY ORDER LOCATED IN 'Al-buhayrat'
339 •  SELECT COUNT(TrackingNum) AS count
340    FROM shippingOrder
341    GROUP BY location
342    HAVING location = 'Al-buhayrat';
343
```

100% 1:45

Result Grid

count
▶ 2

```
344    -- INNER JOIN BETWEEN shippingOrder TABLE AND weightPrice TABLE
345    -- TO LIST THE TrackingNum, location, weight AND ITS shippingPrice
346 •  SELECT TrackingNum, location, w.weight, shippingPrice
347    FROM shippingOrder s, weightPrice w
348    WHERE s.weight = w.weight;
349
```

	TrackingNum	location	weight	shippingPrice
▶	AM6987654212	Al-nawwariyah	6	40
◀	AY1223456789	Al-buhayrat	2	20
◀	BN9842135678	Al-zahir	6	40
◀	HJ1974368346	Al-shoqiyah	3	25
◀	KW1234457890	Al-buhayrat	7	45
◀	SA1234567890	Al-zahir	4	30

SQL DEFINITION COMMANDS IN MYSQL WORKBENCH [JOIN, SUBQUERY, HAVING, WHERE AND GROUP BY COMMANDS]

```
415 -- SUBQUERIES & JOIN -----
416 -- View the information of the most recent customer order with the order number
417 • SELECT O.OrederNum, C.Fname, C.email
418   from `Order` As O
419   INNER JOIN Customer As C
420   On O.CustomerID = C.CustomerID
421   Where O.date = (Select MAX(date) from `Order`);

422 -- View the information of the oldest customer request with the order number
423 • SELECT O.OrederNum, C.Fname, C.email
424   from `Order` As O
425   INNER JOIN Customer As C
426   On O.CustomerID = C.CustomerID
427   Where O.date = (Select MIN(date) from `Order`);

434 -- View Order details -----
435 • Select O.OrederNum, O.date,C.Fname, C.email, S.weight, S.TrackingNum
436   from `Order` As O
437   INNER join Customer As C
438   On O.CustomerID = C.CustomerID
439   INNER join shippingOrder As S
440   ON( S.TrackingNum =  O.TrackingNum)
441   order by O.OrederNum ;
```

	OrederNum	Fname	email
▶	6	Danah	Danah@gmail.com

	OrederNum	Fname	email
▶	3	Reem	Reem@gmail.com

	OrederNum	date	Fname	email	weight	TrackingNum
▶	1	2022-03-12	Danah	Danah@gmail.com	2	AY1223456789
	2	2021-08-25	Ahmed	Ahmed@gmail.com	6	AM6987654212
	3	2021-05-08	Reem	Reem@gmail.com	4	SA1234567890
	4	2022-06-03	Badr	Badr@gmail.com	3	HJ1974368346
	5	2022-09-22	Sarah	Sarah@gmail.com	6	BN9842135678
	6	2023-04-04	Danah	Danah@gmail.com	7	KW1234457890

SQL DEFINITION COMMANDS IN MYSQL WORKBENCH [JOIN, SUBQUERY, HAVING, WHERE AND GROUP BY COMMANDS]

```
436 -- Show the orders order by the quantity of the product -----
437 • SELECT O.OrederNum, P.Name as `Product name` ,C.quantity
438   from `Order` As O
439   INNER JOIN Productdata As P
440   INNER JOIN ContainsProducts As C
441   On P.ProductNum = C.ProductNum
442   On O.OrederNum = C.OrederNum
443   Order by C.quantity ;
...
```

	OrederNum	Product name	quantity
▶	2	Apple Watch	1
	3	iphone 14	4
	6	iphone 14	5
	4	Apple Watch	5
	5	iphone 14	6
	6	iphone 14	8
	6	Samsung Galaxy Buds Pro	9
	1	Samsung Galaxy Buds Pro	12
	5	Apple Watch	18
	2	Apple Watch	20
	4	iphone 14	22
	3	iphone 14	36

```
237 -- Calculate derived attribute (#Seller)
238 • SELECT COUNT(`Seller`.`sellerID`) AS NumberOfSeller
239   FROM `SellingPlatform2`.`Seller`;
240
241 •   SELECT *
242     FROM seller;
243 --- -----
```

NumberOfSeller
▶ 6

Displaying the names of sellers, their identified numbers, who have a Google account and have a quantity of more than or equal to 20 products

```
Limit to 1000 rows
-- Subqueries
462 • SELECT 'Seller'.sellerID, CONCAT('Fname', ' ', 'Lname') AS SellerFullName
463   FROM `SellingPlatform2`.`Seller`,
464        `SellingPlatform2`.`Product`
465   WHERE ('Seller'.sellerID = 'Product'.SellerId) AND 'email' IN (  SELECT 'email'
466     FROM `SellingPlatform2`.`Seller`
467     WHERE ('email' LIKE ('%@gmail.com')) AND 'ProductNum' IN (    SELECT 'ProductNum'
468       FROM `SellingPlatform2`.`Product`
469       WHERE ('Quantity' >= 20)      ));
```

sellerID	SellerFullName
66779984	Abdullah Fahad
11203987	Faisal Jaber