Title (JewelryRetail, JEMS-Group 7, CPSC 332 , 5/16/21, Joseph Nasr, Moses Merugu, Eddie Rayo, Shadi Elkhatib)

Introduction:

Our client named *The Perfect Jem* asked our company JEMS Inc. to design them a database for their imperfect and imitation jewelry store. The capabilities of the Database range from tracking inventory, storing customer's personal and payment information, and recording orders. With regards to products the database stores product and supplier information. In addition, the database keeps track of which customers are members, what discount they receive on their purchase, and what products are exclusive to them.

Database Design Process

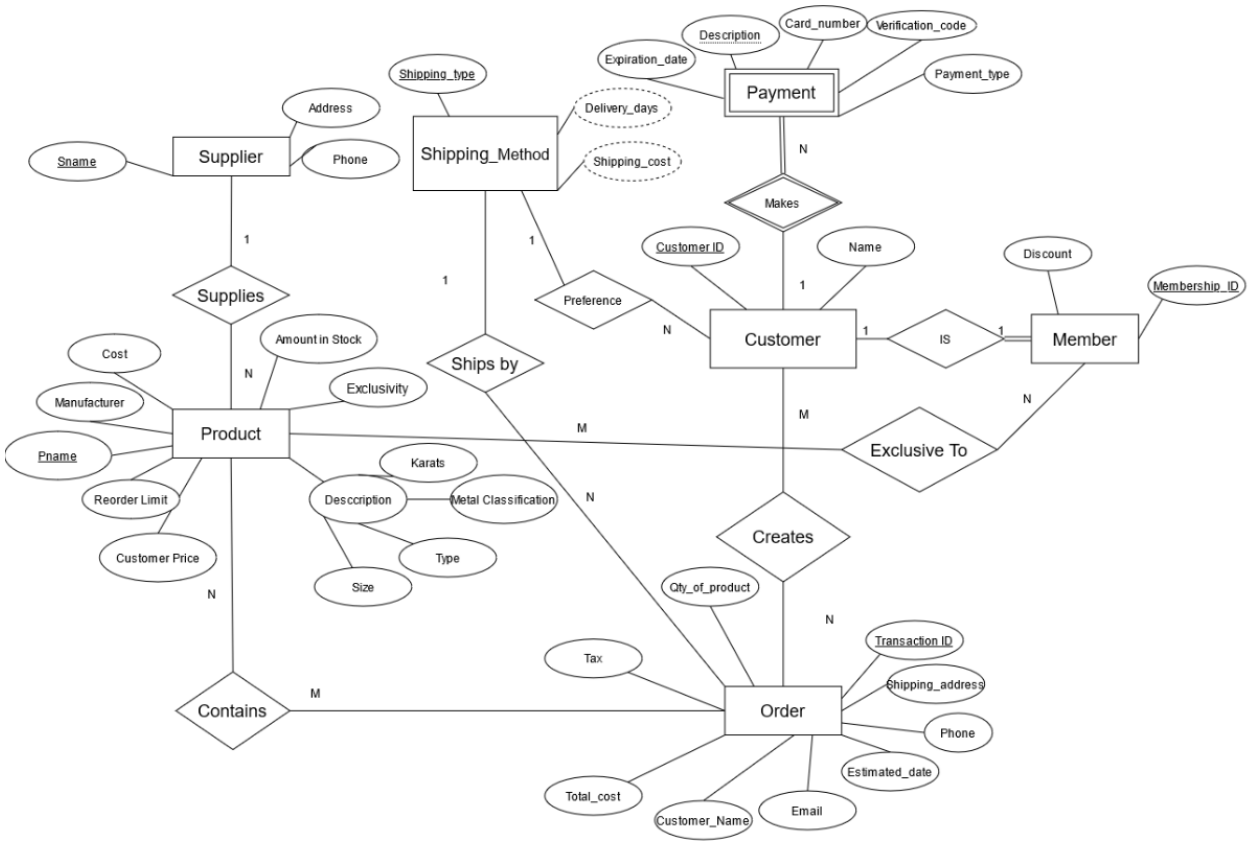
Requirements

1. The online store carries many different jewelry products. We need to track each product by name, manufacturer, description, cost of product, customer price, number of units in stock, as well as any product specific information.
2. Many of the different jewelry products are purchased through different suppliers. We need to track supplier name, address, phone, products that they supply, and cost of product.
3. Customers can order many products in one order, including more than one of the same products. Once an order is complete, we store customer information, including name, shipping address, email address, phone number, product purchase, quantity of product, tax, shipping cost, total cost, and estimated delivery date.
4. Shipping cost is determined by the customers preference. Shipping determines shipping type, cost, and delivery days to ship.
5. We need to collect payment information, including type, number , description, expiration and verification code.
6. A customer can have more than one payment type.
7. Customers can be a member. As a member, customers get special pricing. Members also get access to exclusive products.
8. Every product has a reorder limit. If inventory levels for a product reaches that limit, a procurement request goes to the preferred supplier for that product.
9. Make sure that the Customer ID is the first 2 letter of his first name followed by number. Eg. If the Customer's name is Steve Gates, then his Customer ID can be ST2048.
10. Customers can have more than one shipping address.

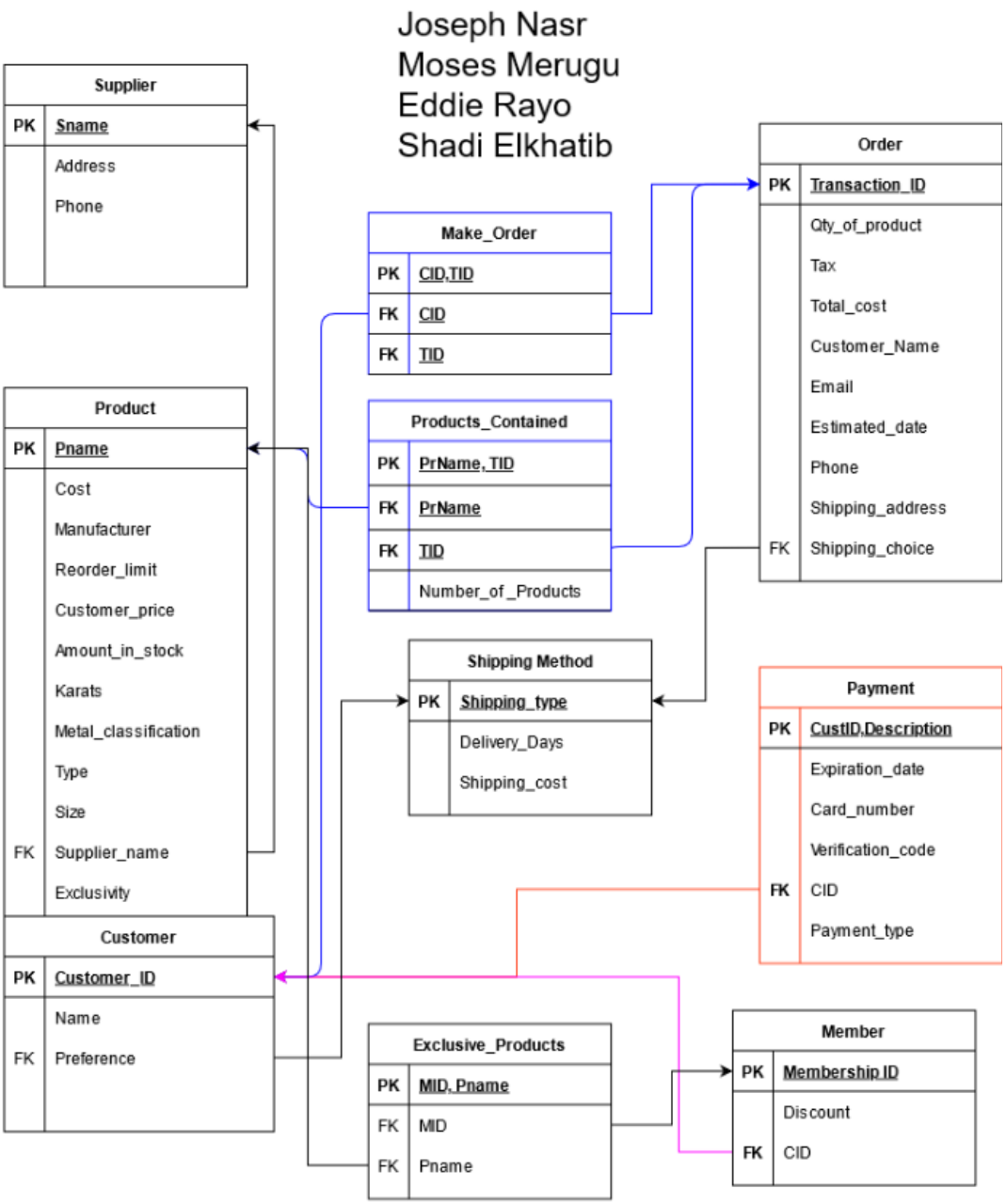
Additional Requirements

1. One SCRIPT to create this database (call it JewelryRetail) with MySQL server.
2. A supplier is recalling the product line Choker Necklace – Extra Small by manufacturer Cheapo Manufacturing. We need to inform all customers that the product is discontinued and ask them to return the product for exchange or a full refund. For this purpose, create a VIEW that finds the names and phone numbers of all customers who purchased the product along with the price they paid for the product.
3. Create a view which has first names and last names of all members who purchased the product NoSparkle Ruby Necklace by Plastic Gem, Inc.
4. Create a view which shows the product name and manufacturer of all products that sold more than 10 units.
5. Modify the view created in Q4 to show the product name and manufacturer of all products that sold more than 10 units and cost more than $50.
6. Create trigger on the Products so that every time a product's inventory level reaches 2, a new entry is made in the Procurement table to order the amount listed for that product. The procurement table will have the following (Hint-The trigger will be on Product table). a. Product Name b. Product Manufacturer c. Action (indicate Order) d. Quantity e. Date of Request
7. Create a script to do the following (Write the script for this) a. If first time backup take backup of all the tables b. If not the first time remove the previous backup tables and take new backups.

ER Diagram



Relational Model



Physical Model

```
ERROR 1146 (42S02): Table 'jems.suppliers' doesn't exist
MariaDB [jems]> describe customer;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| Customer_id | char(6)     | NO   | PRI | NULL    |       |
| Preference  | varchar(20) | NO   | PRI | NULL    |       |
| Name        | varchar(40) | NO   |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
3 rows in set (0.010 sec)

MariaDB [jems]> describe supplier;
+---------+--------------+------+-----+---------+-------+
| Field   | Type         | Null | Key | Default | Extra |
+---------+--------------+------+-----+---------+-------+
| SName   | varchar(20)  | NO   | PRI | NULL    |       |
| Address | varchar(120) | NO   |     | NULL    |       |
| PHONE   | char(16)     | YES  |     | NULL    |       |
+---------+--------------+------+-----+---------+-------+
3 rows in set (0.010 sec)

MariaDB [jems]> describe shipping_method;
+---------------+---------------+------+-----+---------+-------+
| Field         | Type          | Null | Key | Default | Extra |
+---------------+---------------+------+-----+---------+-------+
| Shipping_type | varchar(20)   | NO   | PRI | NULL    |       |
| Delivery_days | int(11)       | NO   |     | NULL    |       |
| Shipping_cost | decimal(10,2) | NO   |     | NULL    |       |
+---------------+---------------+------+-----+---------+-------+
3 rows in set (0.010 sec)
```

```
MariaDB [jems]> describe product;
+----------------------+---------------+------+-----+---------+-------+
| Field                | Type          | Null | Key | Default | Extra |
+----------------------+---------------+------+-----+---------+-------+
| PName                | varchar(40)   | NO   | PRI | NULL    |       |
| Supplier_name        | varchar(20)   | NO   | MUL | NULL    |       |
| Cost                 | decimal(10,2) | NO   |     | NULL    |       |
| Manufacturer         | varchar(40)   | NO   |     | NULL    |       |
| Reorder_limit        | int(11)       | NO   |     | NULL    |       |
| Customer_price       | decimal(10,2) | NO   |     | NULL    |       |
| Amount_in_stock      | int(11)       | NO   |     | NULL    |       |
| Karats               | int(11)       | NO   |     | NULL    |       |
| Metal_classification | varchar(20)   | NO   |     | NULL    |       |
| Type                 | varchar(20)   | NO   |     | NULL    |       |
| Size                 | int(11)       | NO   |     | NULL    |       |
| Exclusivity          | tinyint(1)    | NO   |     | 0       |       |
+----------------------+---------------+------+-----+---------+-------+
12 rows in set (0.011 sec)

MariaDB [jems]> describe member;
+---------------+---------+------+-----+---------+-------+
| Field         | Type    | Null | Key | Default | Extra |
+---------------+---------+------+-----+---------+-------+
| Membership_id | char(6) | NO   | PRI | NULL    |       |
| Discount      | float   | NO   |     | NULL    |       |
| CID           | char(6) | NO   | MUL | NULL    |       |
+---------------+---------+------+-----+---------+-------+
3 rows in set (0.007 sec)

MariaDB [jems]> describe exclusive_products;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| MID   | char(6)     | NO   | PRI | NULL    |       |
| Pname | varchar(40) | NO   | PRI | NULL    |       |
+-------+-------------+------+-----+---------+-------+
2 rows in set (0.007 sec)
```

```
MariaDB [jems]> describe payment;
+-------------------+-------------+------+-----+---------+-------+
| Field             | Type        | Null | Key | Default | Extra |
+-------------------+-------------+------+-----+---------+-------+
| CID               | char(6)     | NO   | PRI | NULL    |       |
| Description       | varchar(30) | NO   | PRI | NULL    |       |
| Expiration_date   | varchar(5)  | NO   |     | NULL    |       |
| Card_number       | char(16)    | NO   |     | NULL    |       |
| Verification_code | char(3)     | NO   |     | NULL    |       |
| Payment_type      | varchar(10) | NO   |     | NULL    |       |
+-------------------+-------------+------+-----+---------+-------+
6 rows in set (0.008 sec)

MariaDB [jems]> describe orders;
+------------------+---------------+------+-----+---------+-------+
| Field            | Type          | Null | Key | Default | Extra |
+------------------+---------------+------+-----+---------+-------+
| Transaction_id   | char(10)      | NO   | PRI | NULL    |       |
| Qty_of_product   | int(11)       | NO   |     | NULL    |       |
| Tax              | decimal(2,2)  | NO   |     | NULL    |       |
| Total_cost       | decimal(10,2) | NO   |     | NULL    |       |
| Customer_name    | varchar(40)   | NO   |     | NULL    |       |
| Email            | varchar(40)   | NO   |     | NULL    |       |
| Estimated_date   | varchar(40)   | NO   |     | NULL    |       |
| Phone            | char(16)      | NO   |     | NULL    |       |
| Shipping_address | varchar(70)   | NO   |     | NULL    |       |
| Shipping_choice  | varchar(40)   | NO   | MUL | NULL    |       |
+------------------+---------------+------+-----+---------+-------+
10 rows in set (0.010 sec)

MariaDB [jems]> describe makes_order;
+-------+----------+------+-----+---------+-------+
| Field | Type     | Null | Key | Default | Extra |
+-------+----------+------+-----+---------+-------+
| CID   | char(6)  | NO   | PRI | NULL    |       |
| TID   | char(10) | NO   | PRI | NULL    |       |
+-------+----------+------+-----+---------+-------+
2 rows in set (0.010 sec)
```

```
MariaDB [jems]> describe products_contained;
+--------------------+-------------+------+-----+---------+-------+
| Field              | Type        | Null | Key | Default | Extra |
+--------------------+-------------+------+-----+---------+-------+
| PrName             | varchar(40) | NO   | PRI | NULL    |       |
| TID                | char(10)    | NO   | PRI | NULL    |       |
| Number_of_products | int(11)     | NO   |     | NULL    |       |
+--------------------+-------------+------+-----+---------+-------+
3 rows in set (0.010 sec)

MariaDB [jems]> describe procurement;
+----------------------+-------------+------+-----+---------+-------+
| Field                | Type        | Null | Key | Default | Extra |
+----------------------+-------------+------+-----+---------+-------+
| Product_Name         | varchar(40) | NO   |     | NULL    |       |
| Product_Manufacturer | varchar(40) | NO   |     | NULL    |       |
| Action               | varchar(40) | NO   |     | NULL    |       |
| Quantity             | int(11)     | NO   |     | NULL    |       |
| Date_of_Request      | date        | NO   |     | NULL    |       |
+----------------------+-------------+------+-----+---------+-------+
5 rows in set (0.010 sec)
```

Application Design

Overview:

The ER diagram we have established consists of six strong entities and one weak entity. Of these entities we have a customer who creates orders, makes payments for those orders, and has a shipping method preference. In addition, a customer can become a member who gets discounts and exclusive merch. When a customer makes an order, that order will contain a product provided by a supplier. At one point our ER diagram looked different than it is now. We originally had Exclusive products as a multivalued attribute of the member entity. However, working through our relational model, it made much more sense to connect Member to Product through an exclusive relationship. In other words, a certain product would be related to members through an Exclusive Products relation. Another issue we ran into was the shipping preference of the customer. We originally had the preference as an attribute of the Shipping_Method Entity and the entity only in relation to Customer and not Order. However, this doesn't make sense because a customer may favor a shipping method, but may choose a different one for a specific order. Thus, we connected Shipping_Method to Order in the sense that every order has a method to ship by and we connected Shipping_Method to Customer to store the preference of the customer. Most of these errors in thinking were caught while making our relational model and putting it into third normal form.

Speaking of our relational model, most of it is self explanatory based on what was said about our ER diagram. However, the two relations that stand out are Make_Order and Products_Contained. Make_Order provides a receipt of who ordered, and what transaction ID they had. Products_Contained keeps a receipt of what products

were in a transaction as well as how many of that product. These two relations serve as a foundation for customers ordering products.

To make our physical model, we simply used ‘create table’, and ‘insert’ commands. In addition, using php my admin we edited some tables that needed more attributes. If we made a mistake in data insertion we would update it. For sample data, we just used different Computer Science department professors as our customers. For products we completely made from our imaginations except for the Choker Necklace and the NoSparkle Ruby Necklace. For Supplier, we just looked up jewelry items in different languages and named the supplier after those translated items. For Orders we had each of our customers buy products. We also had all the Computer Science professors by a Platinum Bracelet, since the professors in the CS department are all best friends.

SQL Queries (Organized based on Table):

```
CREATE TABLE SUPPLIER
(
	SName varchar(20) not null,
	Address varchar(120) not null,
	Phone char(16) not null,
	primary key(SName)
);
INSERT INTO Supplier VALUES ('Cheapo Manufacturing', '89 Elizabeth Court Frankfort, KY 40601', '1-(714)-888-3828');
INSERT INTO Supplier VALUES ('Tekubi Inc.', '93 Chino CA 67980', '1-(909)-487-6345');
INSERT INTO Supplier VALUES ('Halqa Inc.', '72 Placentia CA 45331', '1-(714)-921-8790');
INSERT INTO Supplier VALUES ('Kostbaar Inc.', '54 Fullerton CA 88213', '1-(562)-800-0090');
INSERT INTO Supplier VALUES ('Plastic Gem Inc.', '32 Casper WY 15379', '1-(307)-180-0227');
```

```
CREATE TABLE SHIPPING_METHOD
(
	Shipping_type VARCHAR(20) not null,
	Delivery_days INT not null,
	Shipping_cost DECIMAL (10,2) not null,
	Primary key(Shipping_type)
);
INSERT INTO Shipping_method VALUES ('Express', 2, 3.99);
```

```
INSERT INTO Shipping_method VALUES ('Mid-Tier', 5, 1.99);
INSERT INTO Shipping_method VALUES ('Normal', 7, 0.99);
```

CREATE TABLE **CUSTOMER**

```
(
	Customer_id CHAR(6) not null,
	Preference varchar(20) not null,
	Name varchar(40) Not Null,
	primary key(Customer_id, Preference),
	foreign key(Preference) references Shipping_Method(Shipping_type)
);
INSERT INTO Customer VALUES ('SA1874', 'Express', 'Sam Smith');
INSERT INTO Customer VALUES ('DA0332', 'Mid-Tier', 'Dave Garcia-Gomez');
INSERT INTO Customer VALUES ('PA0121', 'Express', 'Paul Inventado');
INSERT INTO Customer VALUES ('YU0240', 'Normal', 'Yu Bai');
INSERT INTO Customer VALUES ('KE0131', 'Normal', 'Kevin Wortman');
INSERT INTO Customer VALUES ('CS1000', 'Express', 'Christopher Ryu')
INSERT INTO Customer VALUES('AL0323', 'Express', 'Anthony Le');
```

CREATE TABLE **PRODUCT**

```
(
	PName varchar(40) not null,
	Supplier_name varchar(20) not null,
	Cost decimal(10,2) not null,
	Manufacturer varchar(40) not null,
	Reorder_limit int not null,
	Customer_price decimal(10,2) not null,
	Amount_in_stock int not null,
	Karats int not null,
	Metal_classification varchar(20) not null,
	Type varchar(20) not null,
	Size int not null,
	Exclusivity BOOLEAN Not Null DEFAULT False,
	primary key(PName),
	foreign key(Supplier_name) references SUPPLIER(SName)
);
INSERT INTO Product VALUES ('Choker Necklace', 'Cheapo Manufacturing', 49.99, 'Cheapo
Manufacturing', 10, 200.00, 200, 0, 'Gold', 'Necklace', 14, True);
INSERT INTO Product VALUES ('NoSparkle Ruby Necklace', 'Plastic Gem Inc.', 39.99, 'Plastic
Gem Inc.', 10, 79.99, 200, 0, 'Bronze', 'Necklace', 12, False);
```

```
INSERT INTO Product VALUES ('Emerald Ring', 'Halqa Inc.', 89.99, 'Alliance Manufacturing',
2, 359.99, 50, 3, 'Gold', 'Ring', 7, True);
INSERT INTO Product VALUES ('Platinum Bracelet', 'Tekubi Inc.', 39.99, 'Style
Manufacturing', 10, 59.99, 200, 0, 'Platinum', 'Bracelet', 7, False);
INSERT INTO Product VALUES ('Sapphire Earring', 'Kostbaar Inc.', 39.99, 'Fashion
Manufacturing', 2, 89.99, 50, 2, 'Silver', 'Earring', 1, False);
UPDATE Product SET Amount_in_Stock = 2 WHERE Pname = 'Platinum Bracelet';
```

CREATE TABLE **MEMBER**

```
(
	Membership_id CHAR(6) not null,
	Discount float not null,
	CID CHAR(6) not null,
	primary key(Membership_id),
	foreign key(CID) references CUSTOMER(Customer_id)
);
INSERT INTO Member VALUES ('123456', 0.15, 'SA1874');
INSERT INTO Member VALUES('987654', 0.20, 'DA0332');
```

CREATE TABLE **EXCLUSIVE_PRODUCTS**

```
(
	MID CHAR(6) not null,
	Pname varchar(40) Not null,
	primary key(MID, Pname),
	foreign key(MID) references MEMBER(Membership_id),
	foreign key(Pname) references Product(Pname)
);
INSERT INTO Exclusive_products VALUES ('123456', 'Choker Necklace');
INSERT INTO Exclusive_products VALUES ('987654', 'Emerald Ring');
```

CREATE TABLE **PAYMENT**

```
(
	CID CHAR(6) not null,
	Description varchar(30) not null,
	Expiration_date varchar(5) not null,
	Card_number CHAR(16) not null,
	Verification_code CHAR(3) not null,
	Payment_type varchar(10) not null,
	primary key(CID, Description),
	foreign key(CID) references CUSTOMER(Customer_id)
);
```

INSERT INTO Payment VALUES ('SA1874', 'American Express', '11/25', '8371869205728548', '456', 'Credit Card');
INSERT INTO Payment VALUES ('SA1874', 'Visa', '11/25', '9285730291857483', '124', 'Debit Card');
INSERT INTO Payment VALUES ('DA0332', 'Visa', '07/25', '7746385594837295', '897', 'Debit Card');
INSERT INTO Payment VALUES ('PA0121', 'MasterCard', '06/25', '9987456356560012', '901', 'Credit Card');
INSERT INTO Payment VALUES ('YU0240', 'Discover', '03/25', '3286749385068277', '882', 'Debit Card');
INSERT INTO Payment VALUES ('KE0131', 'Chase', '02/25', '4386039286947381', '231', 'Credit Card');
INSERT INTO Payment VALUES ('CS1000', 'Vise', '01/25', '1111222233334444', '909', 'Credit Card');
INSERT INTO Payment VALUES ('AL0323', 'Discover', '08/25', '4897123456618845', '542', 'Debit Card');

CREATE TABLE **ORDERS**
(
	Transaction_id CHAR(10) not null,
	Qty_of_product INT not null,
	Tax decimal(2,2) not null,
	Total_cost decimal(10,2) not null,
	Customer_name varchar(40) not null,
	Email varchar(40) not null,
	Estimated_date varchar(40) not null,
	Phone char(16) not null,
	Shipping_address varchar(40) not null,
	Shipping_choice varchar(40) not null,
	primary key(Transaction_id),
	foreign key (Shipping_choice) references Shipping_method(Shipping_type)
);
INSERT INTO Orders VALUES ('ID-8492857', 2, 0.08, (((79.99+200.00)*0.85)*1.08)+3.99, 'Sam Smith', 'samsmith777@gmail.com', 'Monday May 10th', '1-(714)-395-2964', '1061 N State College Blvd Anaheim CA 92806', 'Express');
INSERT INTO Orders VALUES ('ID-9473216', 3, 0.08, (((359.99+59.99+200)*0.80)*1.08)+1.99, 'Dave Garcia-Gomez', 'dgarciagomez@csu.fullerton.edu', 'Monday May 10th', '1-(714)-738-3879', '2403 E Chapman Ave Fullerton CA 92831', 'Mid-tier');

INSERT INTO Orders VALUES ('ID-4597292', 2, 0.08, ((59.99+200)*1.08)+3.99, 'Paul Inventado', 'pinventado@csu.fullerton.edu', 'Wednesday May 12th', '1-(714)-543-1134', '1061 N State College Blvd Anaheim CA 92806', 'Express');
INSERT INTO Orders VALUES ('ID-1791294', 1, 0.08, ((59.99)*1.08)+0.99, 'Yu Bai', 'ybai@csu.fullerton.edu', 'Thursday May 13th', '1-(714)-444-9834', '2810 E Imperial Hwy Fullerton CA 92835', 'Express');
INSERT INTO Orders VALUES ('ID-9529904', 1, 0.08, (59.99*1.08)+0.99, 'Kevin Wortman', 'kwortman@csu.fullerton.edu', 'Tuesday May 11th', '1-(714)-395-2964', '770 W Chapman Ave Placentia CA 92870', 'Express');
INSERT INTO Orders VALUES ('ID-5670981', 1, 0.08, (59.99*1.08)+3.99, 'Christopher Ryu', 'cryu@csu.fullerton.edu', 'Tuesday May 9th', '1-(714)-790-1234', '800 N State College Blvd, Fullerton, CA 92831', 'Express');
INSERT INTO Orders VALUES ('ID-9812551', 6, 0.08, (59.99*6*1.08)+3.99, 'Anthony Le', 'ale@csu.fullerton.edu', 'Friday May 13th', '1-(714)-482-6645', '805 W State College Blvd, Fullerton, CA 92831', 'Express');

CREATE TABLE **Makes_Order**
(
	CID CHAR(6) NOT NULL,
	TID CHAR(10) NOT NULL,
	primary key(CID, TID),
	foreign key(CID) references Customer(Customer_ID),
	foreign key(TID) references ORDERS(Transaction_id)
);
INSERT INTO Makes_Order VALUES ('SA1874', 'ID-8492857');
INSERT INTO Makes_Order VALUES ('DA0332', 'ID-9473216');
INSERT INTO Makes_Order VALUES ('PA0121', 'ID-4597292');
INSERT INTO Makes_Order VALUES ('YU0240', 'ID-1791294');
INSERT INTO Makes_Order VALUES ('KE0131', 'ID-9529904');
INSERT INTO Makes_Order VALUES ('CS1000', 'ID-5670981');
INSERT INTO Makes_Order VALUES ('AL0323', 'ID-9812551');

CREATE TABLE **Products_Contained**
(
	PrName VARCHAR(40) NOT NULL,
	TID CHAR(10) NOT NULL,
	Num_of_Products INT NOT NULL,
	Primary key(PrName, TID),
	Foreign key(PrName) references PRODUCT(Pname),
	Foreign key(TID) references ORDERS(Transaction_id)

```
);
INSERT INTO Products_Contained VALUES ('Choker Necklace', 'ID-8492857', 1);
INSERT INTO Products_Contained VALUES ('NoSparkle Ruby Necklace', 'ID-8492857', 1);
INSERT INTO Products_Contained VALUES ('Emerald Ring', 'ID-9473216', 1);
INSERT INTO Products_Contained VALUES ('Platinum Bracelet', 'ID-9473216', 1);
INSERT INTO Products_Contained VALUES ('Choker Necklace', 'ID-9473216', 1);
INSERT INTO Products_Contained VALUES ('Platinum Bracelet', 'ID-4597292', 1);
INSERT INTO Products_Contained VALUES ('Sapphire Earring', 'ID-4597292', 1);
INSERT INTO Products_Contained VALUES ('Platinum Bracelet', 'ID-1791294', 1);
INSERT INTO Products_Contained VALUES ('Platinum Bracelet', 'ID-9529904', 1);
INSERT INTO Products_Contained VALUES ('Platinum Bracelet', 'ID-5670981', 1);
INSERT INTO Products_Contained VALUES ('Platinum Bracelet', 'ID-9812551', 6);
```

VIEW COMMANDS FOR ADDITIONAL REQUIREMENTS

1)

See submission files

2)

```
CREATE VIEW Choker_Necklace_Purchasers AS
SELECT Customer_name, Phone, ROUND(Customer_price*(1-Discount)*(1+Tax), 2) AS
Amount_Payed FROM Orders AS O,
Products_Contained AS PC, Product AS P,  Exclusive_Products  AS EP,
Member AS M
WHERE O.Transaction_ID = PC.TID AND PC.PrName = P.Pname  AND P.Pname = EP.Pname
AND
EP.MID = M.Membership_ID AND PC.Prname = 'Choker Necklace'  AND Size = 14;
```

3)

```
CREATE VIEW NoSparkle_Ruby_Necklace_Members AS
SELECT Name FROM Customer AS C, Member AS M, Makes_Order  AS MO,
Products_Contained AS PC WHERE C.Customer_ID = M.CID  AND C.Customer_ID =
MO.CID AND MO.TID = PC.TID AND PC.PrName = 'NoSparkle  Ruby Necklace';
```

4)

```
CREATE VIEW Products_With_GT10_Sold AS
SELECT Pname, Manufacturer FROM Product AS P, Products_contained  AS PC WHERE
P.Pname = PC.PrName GROUP BY Pname HAVING SUM(Number_of_products) > 10;
```

5) ONLY the Altered one will show up in database

ALTER VIEW Product_With_GT10_Sold AS SELECT Pname, Manufacturer FROM Product AS P, Products_contained AS PC WHERE P.Pname = PC.PrName AND Customer_price > 50 GROUP BY Pname HAVING COUNT(*) > 10;

```
MariaDB [jems]> Select * from Products_With_GT10_Sold;
+-------------------+---------------------+------------+
| Pname             | Manufacturer        | UNITS_SOLD |
+-------------------+---------------------+------------+
| Platinum Bracelet | Style Manufacturing |         11 |
+-------------------+---------------------+------------+
1 row in set (0.000 sec)

MariaDB [jems]> ALTER VIEW Products_With_GT10_Sold AS
    -> SELECT Pname, Manufacturer, Customer_price, SUM(Number_of_products) FROM Product AS P, Products_contained AS
PC WHERE P.Pname = PC.PrName AND Customer_price > 50 GROUP BY Pname HAVING SUM(Number_of_products) > 10;
Query OK, 0 rows affected (0.006 sec)

MariaDB [jems]> Select * from Products_With_GT10_Sold;
+-------------------+---------------------+----------------+------------------------+
| Pname             | Manufacturer        | Customer_price | SUM(Number_of_products) |
+-------------------+---------------------+----------------+------------------------+
| Platinum Bracelet | Style Manufacturing |          59.99 |                     11 |
+-------------------+---------------------+----------------+------------------------+
1 row in set (0.001 sec)
```

6)

```
CREATE TABLE Procurement
(
Product_Name VARCHAR(40) NOT NULL,
Product_Manufacturer varchar(40) not null,
Action varchar(40) NOT NULL,
Quantity int NOT NULL,
Date_of_Request DATE NOT NULL
);

DELIMITER //

CREATE TRIGGER Below_2
AFTER UPDATE ON Product
FOR EACH ROW
IF (NEW.Amount_in_Stock = 2 ) THEN
INSERT INTO Procurement VALUES ( OLD.Product.Pname,
OLD.Product.Manufacturer,
'Order',
OLD.Product.Reorder_limit,
CURDATE());
END IF;
 //
DELIMITER;
```

7) Comments are in script to describe what it does

<script>

```
function backUpDB()
{
	// this will output a back up file called JEMS_BUP.sql which will be then used to restore
if needed
	<?php
	$hostName = "localhost";
	$userName = "root";
	$password = "";
	$dbName = "JEMS";

	// this is the directory where the backup file will be downloaded to
	define("BACKUP_PATH", "C:/xampp/htdocs");
	// sqldump command to create a backup of the database JEMS
	$cmd = "C:\\xampp\\mysql\\bin\\mysqldump -u root JEMS > JEMS_BUP.sql";
	exec($cmd);
?>
}

function restoreDB()
{
	<?php
		$hostName = "localhost";
		$userName = "root";
		$password = "";
		$dbName = "JEMS";

		define("BACKUP_PATH", "C:/xampp/htdocs");
		// sql command to restore the database JEMS based on the backup file after the
	‘<’
		$cmd = "C:\\xampp\\mysql\\bin\\mysql -u root JEMS < JEMS_BUP.sql";
		exec($cmd);
	?>
}
</script>
```

Final Product

Homepage:

1) Suppliers Page:

2) Products greater than 10 Sold Page:

Welcome to JEMS INC

FIND THE BEST DEALS FOR ALL YOUR JEWERLY NEEDS

Joseph Nasr, Eddie Rayo, Moses Merugus, Shadi Elkhatib

Manufacturers below have sold their product more than 10 times.

Product Name	Manufacturer	Units Sold
Platinum Bracelet	Style Manufacturing	11

Go Back

3) Choker Necklace recall Page:

Welcome to JEMS INC

FIND THE BEST DEALS FOR ALL YOUR JEWERLY NEEDS

Joseph Nasr, Eddie Rayo, Moses Merugus, Shadi Elkhatib

Below are the Customer and their information to contact for the choker necklace recalll.

Customer name	Phone Number	Amount Payed
Sam Smith	1-(714)-395-2964	187.00
Dave Garcia-Gomez	1-(714)-738-3879	183.60

Go Back

4) Who purchased the NoSparkle Ruby Necklace Page:

Welcome to JEMS INC

FIND THE BEST DEALS FOR ALL YOUR JEWERLY NEEDS

Joseph Nasr, Eddie Rayo, Moses Merugus, Shadi Elkhatib

Below are the customers who have purchased the NoSparkle Ruby Necklace.

Customer name
Sam Smith

Go Back

5) Trigger, Procurement Page:

Welcome to JEMS INC

FIND THE BEST DEALS FOR ALL YOUR JEWERLY NEEDS

Joseph Nasr, Eddie Rayo, Moses Merugus, Shadi Elkhatib

Below is the trigger when product is too low.

Product	Product Manufacturer	Action	Quantity	Date Of Request
Platinum Bracelet	Style Manufacturing	Order	10	2021-05-16
Platinum Bracelet	Style Manufacturing	Order	10	2021-05-16
Platinum Bracelet	Style Manufacturing	Order	10	2021-05-16

Go Back

6) Admin Page:

Summary

The website for *The Perfect JEM* has a friendly user interface design with a home web page that consists of 6 buttons that allows the user to click and access the corresponding button's web page. Each of the web pages has a “go back” feature that enables the user to go back to the home web page. The first page shows a list of suppliers that supply to *The Perfect JEM.* The second page displays the information of customers who have purchased the, now recalled, Choker Necklace. The information of these customers includes their names, phone number, and the price they paid for the Choker Necklace. The third page shows the product name and respected manufacturer of products that have sold more than ten units as well as the number of units sold. The fourth page shows the names of those who have purchased the NoSparkle Ruby Necklace. The fifth page shows a procurement table, which is updated through a trigger, that displays the product name, manufacturer, action, quantity, and date of request of a product whose inventory level reaches two. Once the inventory level of a product reaches two, an entry is made in the procurement table to order the amount listed for that product. The sixth page is where the admin can choose to either back up the current database base or restore it.

JEMS Inc. authorizes this Database and website to the client The Perfect Jem for whatever requisite they aspire to fulfill.

Partner analysis: The group worked really well together. Whenever we needed to meet everyone would show up and participate. Each of us did our fair share of work. We

would usually meet on Wednesday or Friday night to work on the project. Meetings were very productive and we got a lot done in each meeting. Kudos to Eddy, who was host of our database and main creator of our website.

Electronic Signature:
Joseph Nasr: 25%
Eduardo Rayo: 25%
Moses Merugu: 25%
Shadi Elkhatib: 25%