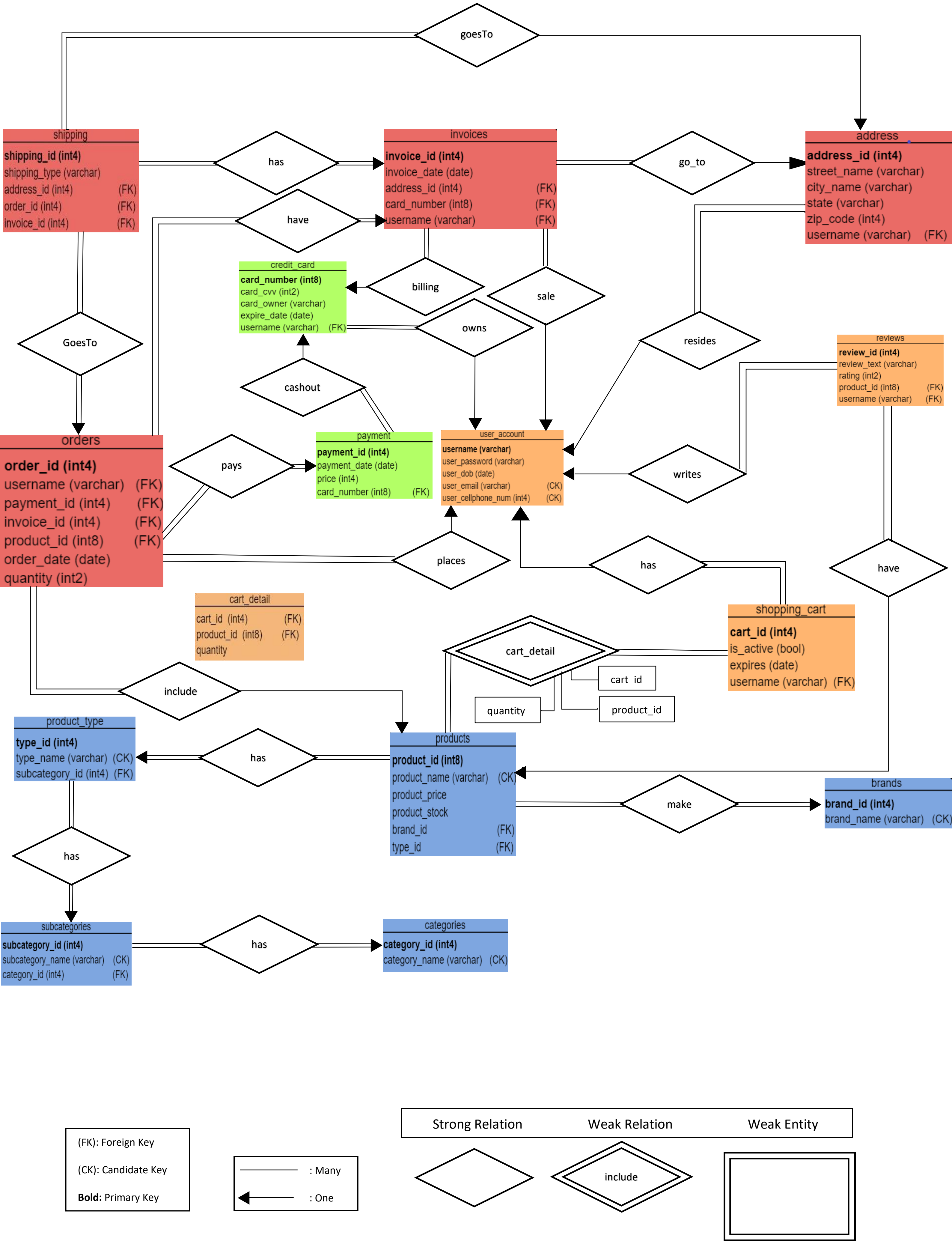
Şevket Kağan ASLAN - 35612186002

Ali Doğukan Dulupçu – 45739307334

Table Specifications

- A user account is stored by its username, password, users birth date, email and cellphone number and each user are identified by their username.
- A credit card is stored by its number, three-digit security pin, holders name, expire date and the user that has used the card and each card is identified by its card number.
- A payment is stored by its date, total amount paid, the credit card which performed the payment and each payment is identified by a unique ID number.
- An address is stored by street name, city name, state, zip code, username who is residing in that address and each address is distinguished by a unique ID number.
- An invoice is stored by its date, the address and username who made the purchase, card number which performed the purchase and a unique id number to identify each invoice.
- An order is stored by the user who placed the order, the payment which paid for the order, invoice ID which can include more than one order, ID of the ordered product, date of the order, amount of the ordered product and each order is identified by a unique number.
- A product is stored by its name, price, stock, brand ID, type ID and a unique product ID which identifies each product and every product’s ID number include information about its category, subcategory and type.
- A shipping is stored by its type, ID of the order, the address which the order needs to be shipped, ID of an invoice and each shipping is identified by a unique number.
- Reviews are stored by the texts and ratings between 0-5 which are written by a user, ID of the product which the review belong to, name of the user which wrote a review and each review is identified by a unique ID number.
- Shopping carts are stored by a Boolean value which indicate that they are active or not, date of expire, name of the user which the cart belongs to and they are identified by a unique ID number.
- Cart contents are stored by ID of the shopping carts and products, quantity of the product added to the cart and they do not have an identifying attribute.
- Brands are stored by their names which are unique and identified by an ID number which is also unique.
- Categories are stored by their unique names and they are identified by a unique ID number.
- Subcategories are stored by their unique names, ID of the category which they belong to and they are identified by a unique ID number.
- Product types are stored by their unique names, ID of the subcategory which they belong to and they are identified by a unique ID number.



(FK): Foreign Key

(CK): Candidate Key

Bold: Primary Key

: Many

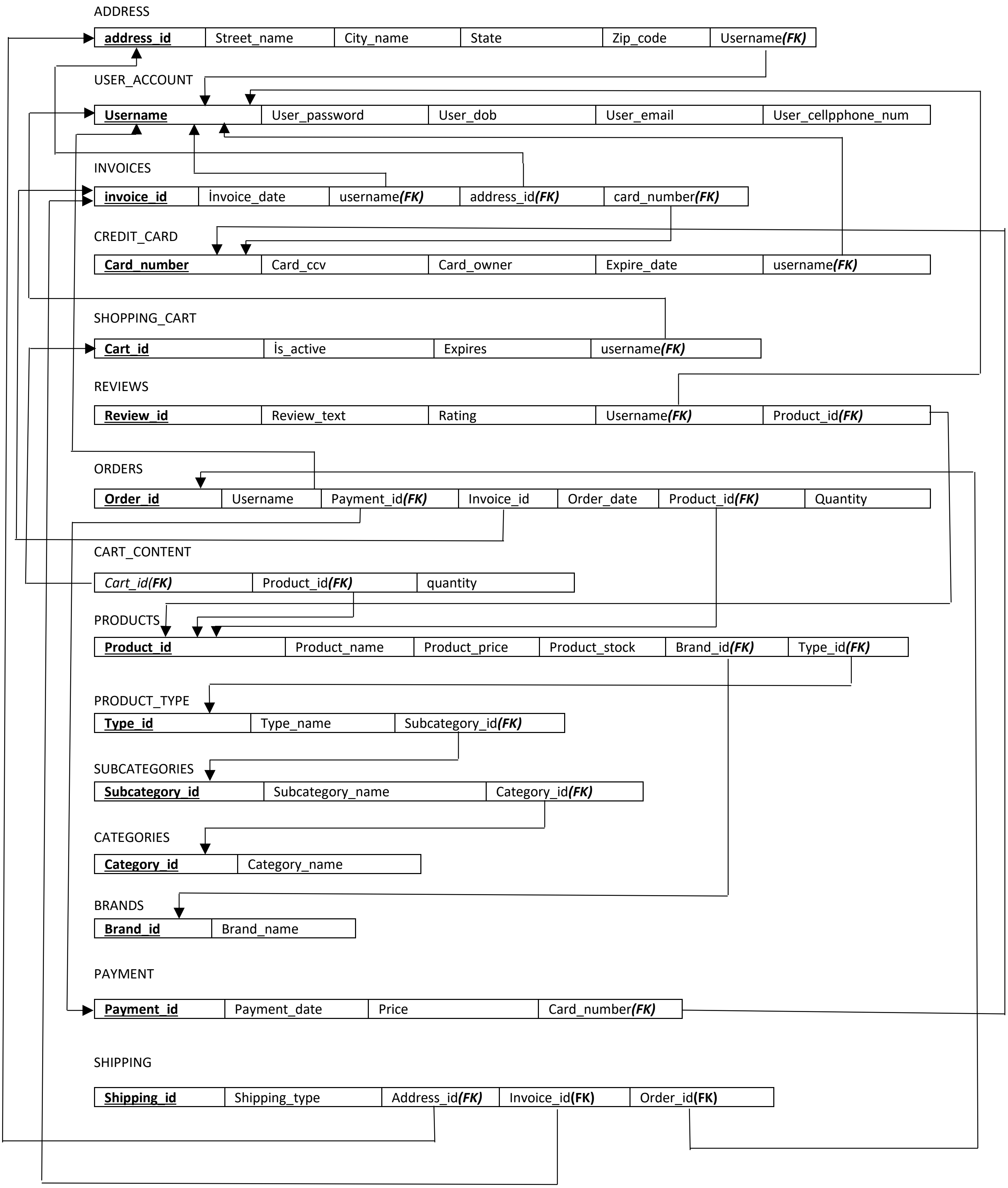
: One

Strong Relation

Weak Relation

Weak Entity

include



ADDRESS

address_id	Street_name	City_name	State	Zip_code	Username*(FK)*

USER_ACCOUNT

Username	User_password	User_dob	User_email	User_cellpphone_num

INVOICES

invoice_id	İnvoice_date	username*(FK)*	address_id*(FK)*	card_number*(FK)*

CREDIT_CARD

Card_number	Card_ccv	Card_owner	Expire_date	username*(FK)*

SHOPPING_CART

Cart_id	İs_active	Expires	username*(FK)*

REVIEWS

Review_id	Review_text	Rating	Username*(FK)*	Product_id*(FK)*

ORDERS

Order_id	Username	Payment_id*(FK)*	Invoice_id	Order_date	Product_id*(FK)*	Quantity

CART_CONTENT

Cart_id(FK)	Product_id*(FK)*	quantity

PRODUCTS

Product_id	Product_name	Product_price	Product_stock	Brand_id*(FK)*	Type_id*(FK)*

PRODUCT_TYPE

Type_id	Type_name	Subcategory_id*(FK)*

SUBCATEGORIES

Subcategory_id	Subcategory_name	Category_id*(FK)*

CATEGORIES

Category_id	Category_name

BRANDS

Brand_id	Brand_name

PAYMENT

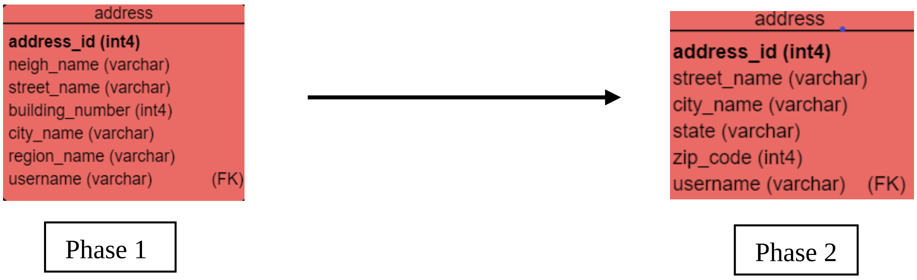
Payment_id	Payment_date	Price	Card_number*(FK)*

SHIPPING

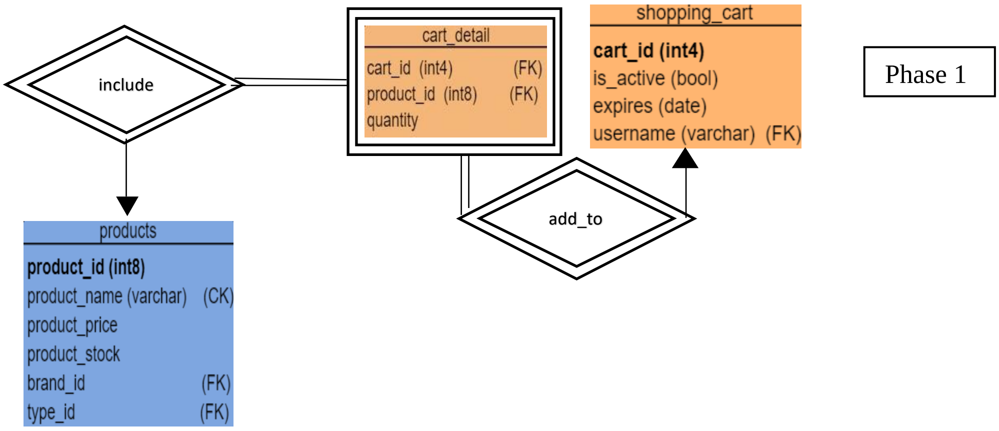
Shipping_id	Shipping_type	Address_id*(FK)*	Invoice_id**(FK)**	Order_id**(FK)**

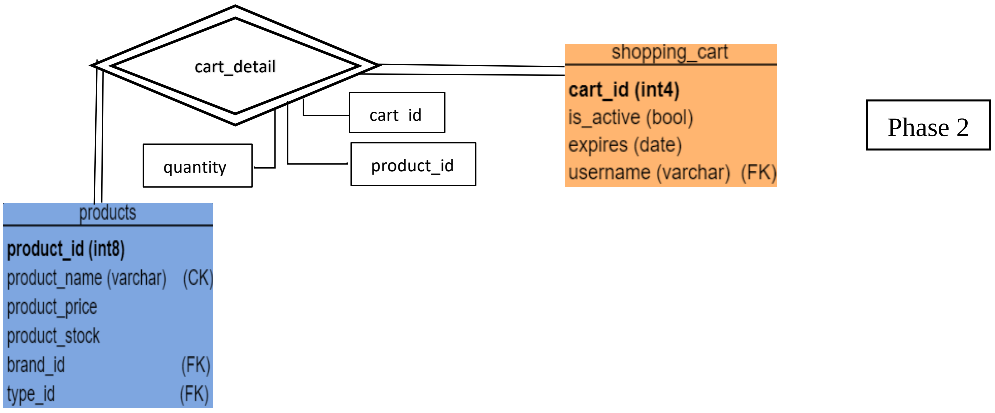
Changes Since Phase 1

- In this project. We as a team believe that we managed to create a working database with very small problems during phase 1. In the second phase we made 2 changes in our project in total. First one is in the address table. To make inserting data into the addresses table a little easier. Changes are shown below:



- Last change in our project is at the ER diagram section. In our database we planned to have a many-to-many relationship between products and shopping_cart tables. Therefore, we created another table called cart_detail to make the relation happen. However, in the ER diagram we have shown cart_detail as a weak entity and it caused us to lose a few points. However, in the phase 2 we changed it so cart_detail is shown as a weak relation.





Java Application Summary

In the Java application for our database we created the connection between school server and Java. After that we opened a statement to send our SQL queries to our database. After that we checked if the app connected or not. To create the menu, we used an infinite loop with for clause that has 4 tasks and an exit. For printing the data from our database, we opened a Result Set to list the data. We have 3 tables to list so we opened 3 Result Sets (rst, rst1, rst2). For insertion ,deletion and update, we used executeUpdate method and to see how many deletion, insertion and update, first we created a String includes the queries (UPDATE,INSERT,DELETE) and after that we run the executeUpdate method to see how many data updated, inserted or deleted to break the infinite loop, we used break clause. For the exceptions we used try and catch clauses. Lastly we used finally block to close all connections, result sets, statements and scanners and with the close method to exit from the program completely.

We chose 3 tables to use in our Java application. For the insertions we chose products table, for the update we chose the cart_content table and lastly we chose orders table for delete operation. To end it all we have created a viewing system to list each separately to check the changes we made in our database though our Java application.