HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY
INSTITUTE OF INTERNATIONAL EDUCATION

FINAL ASSIGNMENT REPORT

BUILDING A WATCH SALES WEBSITE DATABASE

Major: **Information Technology**

Course code: **COS101**

Course name: **Advanced Database**

Lecturer’s name: **TS.Nguyễn Thị Hải Bình**

Student’s name:

1. Huỳnh Minh Toàn	ID: 1811061845	Class: 18DTHQA1
2. Trang Gia Huy	ID: 1811061125	Class: 18DTHQA1
3. Hồ Trịnh Mẫn	ID:1811063282	Class: 18DTHQA1

Ho Chi Minh City

<2021>

SOCIALIST REPUBLIC OF VIETNAM

Independence – Freedom - Happiness

COMMENTS OF THE LECTURER

Student's full name:

(1)... **ID:** ....

(2)... **ID:** ....

(3)... **ID:** ....

(4)... **ID:** ....

(5)... **ID:** ....

Lecturer's comments:

....

....

....

....

....

....

....

....

HCMC, ………. (date)

Please sign here

Lecturer's full name

CONTENT

CHAPTER 1: INTRODUCTION

- Due to the rapidly increasing demand for clothing and housing of people. Especially the need to wear is very necessary to solve that. There are many websites that sell clothes for themselves, in which watches are indispensable.
- To have a website that sells the best watches, in addition to a beautiful interface, good product quality, good brand, we also need a good database to manage all mail. From there, a database topic about the website selling watches was born.

CHAPTER 2: USER REQUIREMENT DETERMINATION

2.1. User

- Customers after registering for an account will have a customer code, customer name, phone number, customer address and email. Customers can edit their information. And they can view - choose products, edit their own shopping cart, choose a payment method.

2.2. Admin Management

- Admins have their own admin code, admin name.
- Manage the number of products sold and the number of products in stock.
- Update more information and import products.
- Edit the selling price, warranty period, product posting date, product location in the directory.

2.3. Product Management

- Manage product codes, specific product names, import prices, selling prices, product information, product types (brand, material, version, logo, accompanying product). Number of products.

2.4. Order Management

- The necessary information has been provided by the customer , such as name, phone number, address. Order with order number and name, and Payment Method. Delivery time (date of order - date of delivery). Orders are moderated by the admin.

2.5. Cart Management

- Each basket has a cart code. Contains products added by the customer. In the cart, there is the total quantity of the product and the total amount of the product. How to calculate the amount of 1 product in the cart by quantity * unit price.

2.6 Non-Functional Requirement

- Regularly update new products
- Customers have a product review card
- Login, can add, delete, edit, manage employees
- Layout interface is easy to see, easy to use
- Show out of stock products (admin page)
- Show top selling products
- Store products after sale for convenient warranty and repair
- Filter products by brand, price, material, color, features
- Ability to interact between buyers and sellers
- High information security (anti-hacker)
- Information about promotions is also placed in an easy-to-see position to help customers feel confident and stimulate purchases.

CHAPTER 3: CONCEPTUAL DATA MODELLING

3.1. Identity entity set

- After providing the customer with the necessary data in the watch sales web database, we created the identity entity set table.

Entity set name	**Attribute**	**Description**
Product (Strong entity set)	Id_Pro (Primary key) Name_pro Status Price Pro_type Quantity_pro	Product code Product name Product status The selling price of the product. Product type Number of products

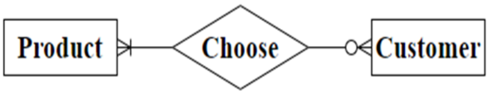
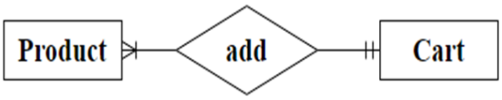
Customer (Strong entity set)	Id_Cus(Primary key) Name_Cus Phone_number Address Email	Customer code Customer name Customer voice amount Customer address Customer email
Cart (Strong entity set)	Id_Card(Primary key) Total Quantity	Cart code Total product price in cart Number of products in carts
Bill (Weak entity set)	Id_bill (Primary key) Delivery_date Order_date Bill_name	Bill code Delivery date Order date Bill's name
Admin (Strong entity set)	Id_ad (Primary key) Name_ad Email_ad	Staff code Employee name Employee Email

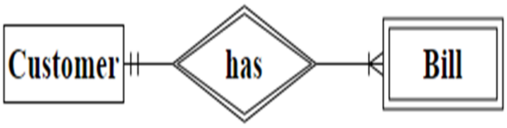
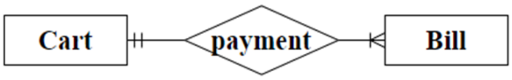
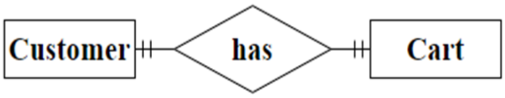
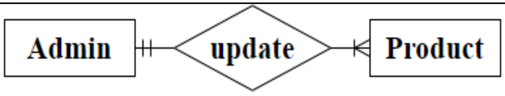
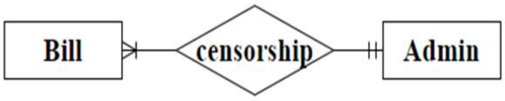
- **Note:**

- In this section, we will define the Strong entity set, the weak entity set is on the tables to be managed mentioned in the Requirement summary section, and the properties of that Entity set.
- The Description is the part that explains the meanings of attributes when we have abbreviated those attributes to be more confident in the design of the database.

3.2. Identify relationship sets and cardinality constraints

- From the identity entity set table above, we continue to build identify relationship sets and cardinality constraints.

Relationship set name	Attribute	Description	Cardinality constraint
Choose		And they can view - choose products	
add		Selected products are added to a shopping cart	

has		The necessary information has been provided by the customer	
payment	Payment methods date	Customers choose a payment method.	
has		Customers edit your own shopping cart	
update		Admin updates more information and imports products.	
censorship		Bill are censorship by admin	

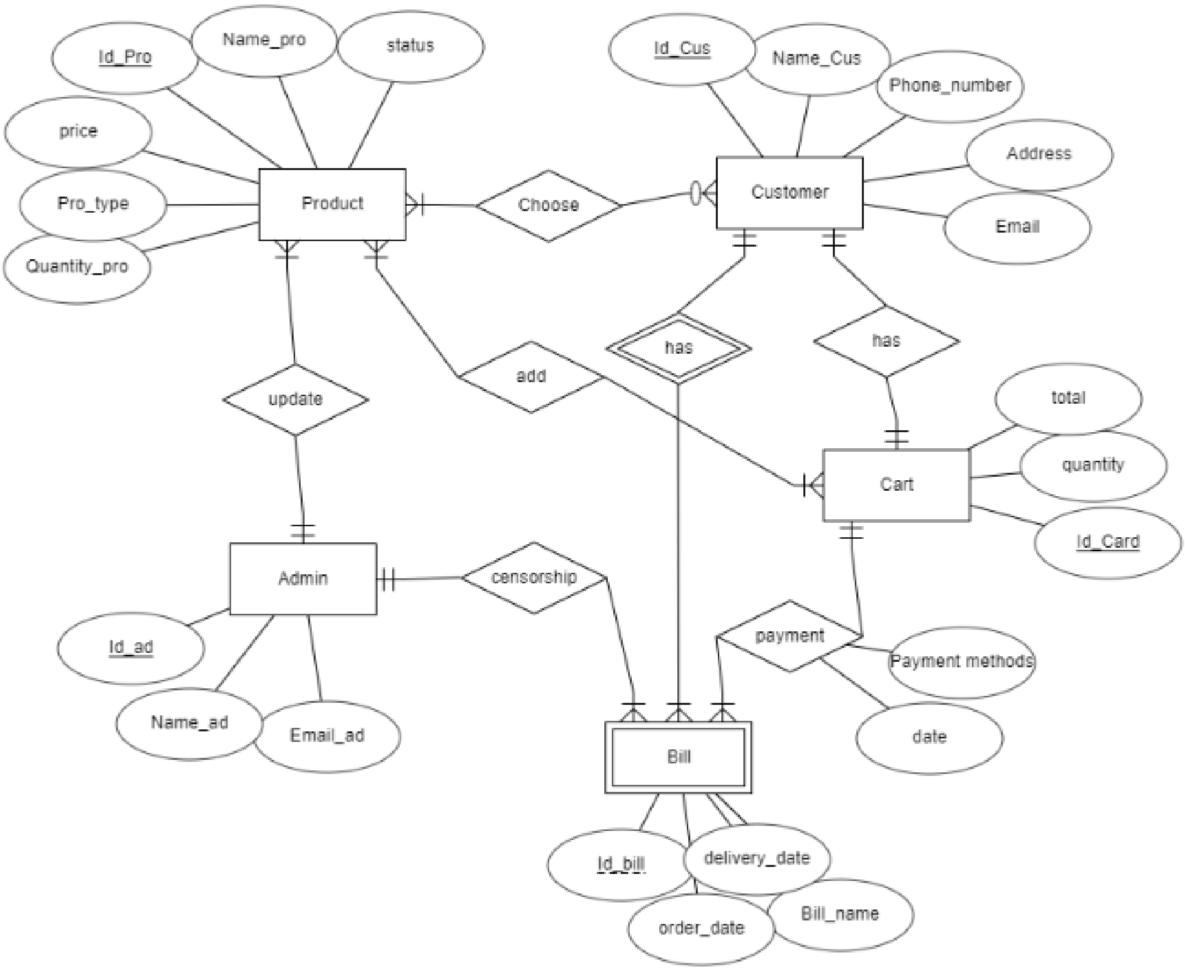
- **Note**

- This section defines the relationships between entity sets based on the meaning of the tables we manage.
- We can choose a name for the relationship, but it should be readable and understandable. Or the relationship name is listed in the Requirement summary section.
- The relationship between Product and Customer is Choose. A product is selected by zero or more customers and a customer selects one or more products
- The relationship between Product and Cart is Add. A Product is added to one and only one Cart and a Cart contains one or more Products
- The relationship between Customer and Bill is Has. A Customer has one or more Bills and a Bill has one and only one Customer
- The relationship between Cart and Bill is Payment. A shopping cart payment one or more Bills and a Bill is payment by one and only one Cart
- The relationship between Customer and Cart is Has. A customer has one and only one shopping cart and vice versa
- The relationship between Admin and Product is Update. Admin account updates one or more products and products updates by one and only admin account

- The relationship between Bill and Admin is Censorship. Bill is censored by one and only admin and admin censorship one or more bills

3.3. Entity relationship diagram

- Here is the entity relationship diagram table.



CHAPTER 4: RELATIONAL DATABASE DESIGN

4.1. ERD to relational schemas

4.1.1Convert entity sets:

- Admin **(<u>Id_ad</u>** , Name_ad , Email_ad)
- Customer (**<u>Id_Cus</u>**, Name_Cus, Phone_number, Address, Email)
- Product (**<u>Id_Pro</u>** , Name_pro, Status, Price, Pro_type, Quantity_pro)
- Card (**<u>Id_Card</u>** , Total, Quantity)
- Bill (**<u>Id_bill</u>** , **<u>Id_cus</u>**,Delivery_date , Order_date , Bill_name)

- **Note**

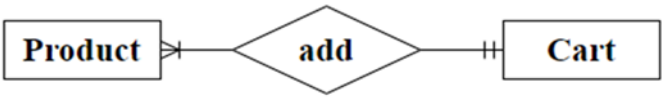
- In this section, we will record the properties of the Strong entity sets, and it is easier to see the Entity reality ship diagram.
- For Bill, when it is a weak entity set, there will be an additional Id_cus as the primary key field because Customer is identifying, because if there is no customer, there is no bill.

4.1.2 Convert relationship sets:

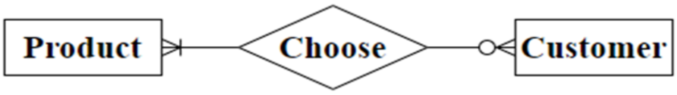
- Convert Choose relationship set: Cus_Pro (**Id_Pro** (FK), **Id_Cus** (FK))
- Convert add relationship set : Pro_Cart (**Id_Pro** (FK), **Id_Card** (FK))
- Convert update relationship set : Product (**Id_Pro** , Name_pro , status , quantity_pro , pro_type , price , Id_ad(FK))
- Convert has relationship set: Card (**Id_Cart**, quantity, total, Id_Cus (FK))
- Convert has relationship set: Bill (**Id_bill ,** Bill_name, order_date , delivery_date , Id_Cus (FK))
- Convert payment relationship set:Bill(**Id_bill** , Bill_name , order_date, delivery_date , Id_Card (FK))
- Convert censorship relationship set: Bill(**Id_bill**, Bill_name, order_date, delivery_date, Id_ad (FK))

- **Note:**

- In this section, we define the foreign key attribute based on the relationship, to make it easier to understand, there is an example:



- The relationship in the product subdivision is 1 to many. Then the Product version will have an additional foreign key that is the primary key of the entity set Cart.



- With a many-to-many relationship as above, we create a new copy with a combined name between the two entities, Pro_Cus and its components are the two primary keys of the two entity sets, and it will become the secondary key in the new version.

4.1.3 Finally, we obtain the following relational schemas:

- Cus_Pro (**Id_Pro** (FK), **Id_Cus** (FK))
- Pro_Cart (**Id_Pro** (FK), **Id_Card** (FK))

- Product(**Id_Pro**, Name_pro, status, quantity_pro, pro_type, price, Id_ad(FK))
- Admin **(Id_ad** , Name_ad , Email_ad)
- Customer (**Id_Cus**, Name_Cus, Phone_number, Address, Email)
- Cart (**Id_Cart**, quantity, total, Id_Cus (FK))
- Bill (**Id_bill** , **Id_cus(**FK),Delivery_date , Order_date , Bill_name, Id_Card(FK), Id_ad(FK))

- **Note**

- This section is a summary of what has been done from the two tables above. After converting relationship sets, the entity sets will have additional subkeys.

4.2. Normalization

4.2.1 Find all function dependencies

- Id_Cus --> Name_Cus, Phone_number, Address, Email
- Id_ad --> Name_ad , Email_ad
- Id_Pro --> Name_pro, status, quantity_pro, pro_type, price, Id_ad
- Id_bill , Id_cus --> Delivery_date , Order_date , Bill_name, Id_Card, Id_ad
- Id_Cart --> quantity, total, Id_Cus
- Id_ad --> Name_ad , Email_ad

4.2.2 Check if the database is in 1NF, 2NF, 3NF.

- The database is in 3NF:

+ 1NF is satisfied because all attributes have atomic values.

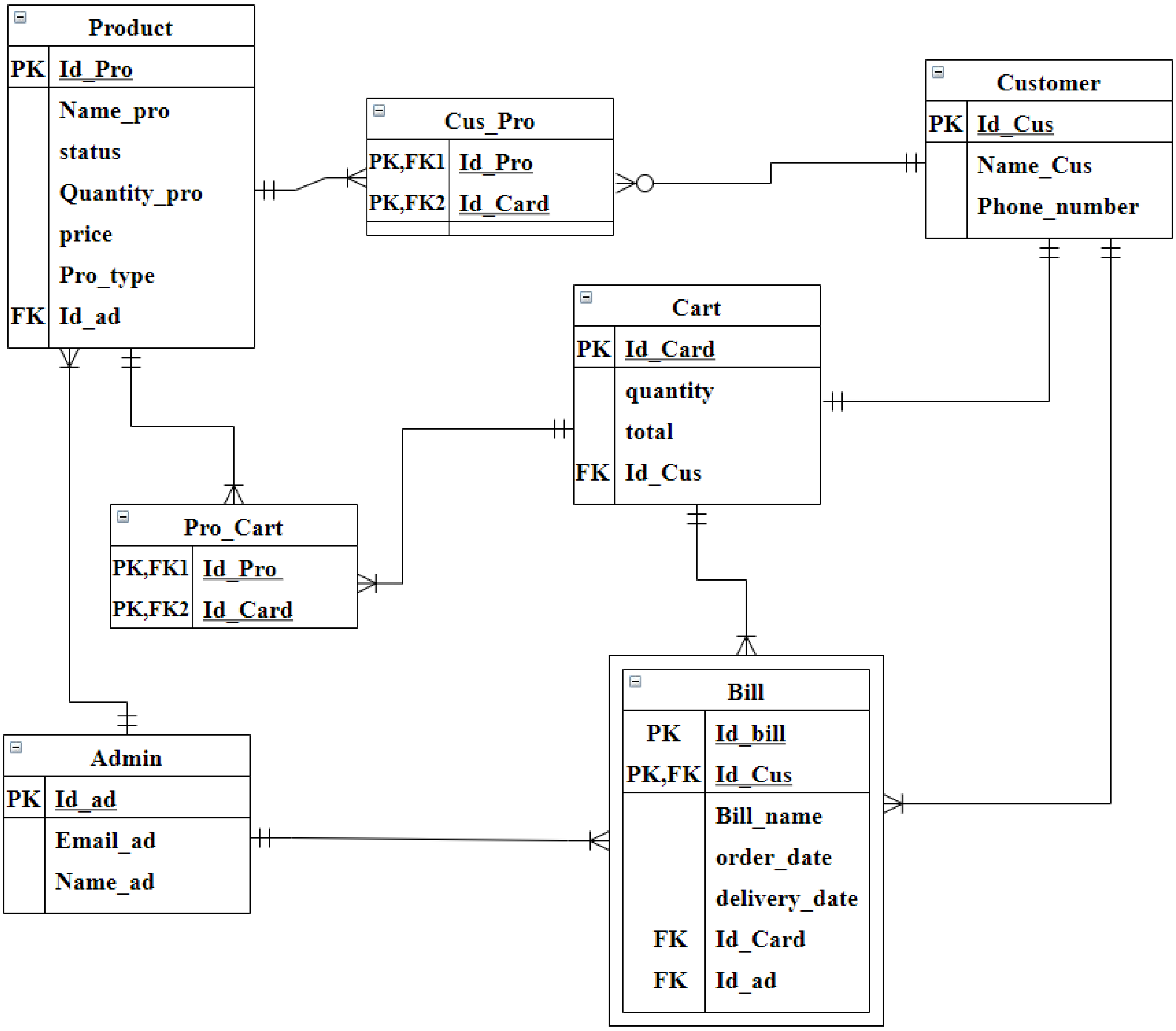
+ 2NF is satisfied because 1NF is satisfied and all non-prime attributes are fully dependent on the key.

+ 3NF is satisfied because 1NF and 2NF are satisfied and no non-prime attributes are transitively dependent on keys.

4.2.3 Normalization: Since the database is in 3NF, the normalization is not necessary.

- Normalization: Since the database is in 3NF, the normalization is not necessary.

4.3 Database diagram



Reference

https://drive.google.com/drive/folders/19w1riha0CUAL_yqeJ2vdIkt1dN4QCavE

https://classroom.google.com/u/1/c/MzkyMDIxNjM2MDc4/m/NDA5NDIzNjQxNjI1/details

https://classroom.google.com/u/1/c/MzkyMDIxNjM2MDc4/m/NDA5NDIzNjQxNjI1/details

https://classroom.google.com/u/1/c/MzkyMDIxNjM2MDc4