ICTDBS502

Assessment 2

ICTDBS502
Design a database

Assessment 2 - TAFE Sunnies

Jina

Date: 21/11/2020

Teacher: Manuela Perez

Table of Contents

Design the database of TAFE Sunnies 3
1. Task 1 – Meeting with client 3
2. Task 2 – Technical requirements 8
2.1 Hardware Requirements 8
2.2 Software Requirements - DBMS Requirements 8
2.3 Functional Requirement 8
2.4 Business Requirement 10
2.5 Security Requirement 11
3. Task 3 – Conceptual model 11
4. Task 4 – Logical data model 14
4.1 Normalisation process 14
4.1.1 Customer 14
4.1.2 Post 15
4.1.3 ShoppingCart 15
4.1.4 Order 16
4.1.5 OrderDetail 16
4.1.6 Product 17
4.1.7 productPhoto 17
4.1.8 Provider 18
4.2 Final ERD 19
4.3 Data Dictionary 20
4.4 Approval to the client 25
5. Task 5 – Business rules and constraints 25
5.1 Business rules 25
5.2 Integrity constraints 26
5.3 Referential integrity constraints 26
5.4 Semantic and other constraints 27
5.5 Indexes 27
5.6 Estimation of the approximate size of the completed database 29
6. User interface 30
6.1 Member registration 30
6.2 Login 31
6.4 Product List 34
6.5 Product Details 35
..... 35
6.7 Invoice Report 38
7. Security 40

8. References..... 41

Design the database of TAFE Sunnies

1. Task 1 – Meeting with client

a) Questions

- **Hardware**
  1. Do you have any request about using a certain Hardware?
     What kind of hardware will you use?
  2. Would you prefer us to use a special platform for the development of your database?
  3. How much storage space do you use physically and online?
  4. Do you have a budget for server – side hardware costs to host server.
  5. How much performance and durability is necessary?
  6. Do you have a special hardware brand or equipment preference?
     - HP, DELL, ASUS
     - CPU (Intel, AMD)
     - RAM Size
     - Hard disk (SATA or SAS type and the size)

  About the space environment to install hardware.
  7. How much space do you have for installation?
  8. If space is small, the number of hardware will be limited.
  9. Does it have proper ventilation?
  10. Improper ventilation can cause the additional cost for the fan or air conditioner facilities to cool off the servers.
  11. Does it have enough power supply?
  12. Does it need to UPS (Uninterruptible Power Supply)?

- **Software**
  1. what computer systems are you using? (browser, computer database, excel, any software package, etc)
  2. Is there any requirement for using the specific operating system?
     For example, Windows NT Server, specific version of linux (Redhat Linux)
  3. Is there any requirement for using the specific application?
     For example: MS SQL Server as a Database Management System, open sourced based applications or commercial based applications.
  4. Do you want to connect the database to any external applications (e.g. outlook)?
  5. What kind of applications will be retrieving posting data to the database, e.g.- Website / app / application?
  6. Do you have a budget for software?

- **Data requirements**
  1. what records do you keep for all your business activities?
  2. How do you keep your records? MS Excel, Access, etc?
  7. What data do you wish to collect from:
     i. Customers
     ii. Employees

iii. Suppliers/manufacturers
iv. Other business partners

8. How do you currently store data relating to your products and do you wish to make any changes to the types of data stored?
9. Where do you see the most data entry mistakes?
10. How would you like us to organize all the stock as the categories of your product? For instance, by material, style, etc
11. What types of data will you be storing for site products?
12. Will you be storing high quality images for your products?
13. Is there anything other than products, member information, worker information, you may want to store on the database. E.G. – business documents?
14. How often will you be cleaning the database, removing old files products, and other information.

- **People – roles, access, etc.**
  1. What are the different groups of users that will access the database and what should each group be able to access?
  2. What will the procedure be for these groups to access the database (e.g. password and usernames)?
  3. Which are the users you want to have on the database and how far do you want their access to be? (permission levels)
  4. What different types of management team will you have and what permissions will each of them have?
  5. Who will be maintaining the database and adding removing old/new items?

- **Processes/business activities**
  1. How wide are you expecting the site to be?
  2. How many users do you estimate will be accessing the database and how much growth do you expect in the coming years?
  3. What is your target number of members, store items?
  4. How many sales have you made since you started?
  5. How many staff are there and what do they do in the company?
  6. What is the invoicing process used for products sold?
  7. Are there any specific business processes that would require data to be labelled by or grouped into stages (e.g. approved, rejected, sent back)?
  8. Will you need to create reports from the stored data (e.g. auditing or stocktake)?
  9. What would you like to know about your costumers, for example, statistics?
  10. What is the information that you are using at the moment to keep track on all your business processes (your stock, what has been sold or returned, orders cancelled and refunds).
  11. What is the workflow to be implemented?
  12. Is there any requirement for design, for example, specific color or design concept, the style of user interface?

- **Business rules**
  1. Do you wish to offer a membership of the company? If yes, would you prefer it to be personal or familiar?
  2. Which benefits would you like a membership to have? (After 3 purchases get a discount or anytime discounts, etc)

3. How would you like users to gain access to their accounts?
4. Do you wish to authenticate the users by a code confirming their emails or mobile numbers?
5. What is your business policy in case of any circumstance? (returning item, cancelling an order, requesting a refund)
6. What is the process involved for any credit/refund requested?
7. Do you offer discounts and specials from time to time with any of your products?
8. Are there any specific business rules that control naming conventions?
9. How often do you have promotions and sales, who will be manipulating the database?

- **Security**
  1. Are you on a LAN, private or public network?
  2. what security systems do you use to protect your products physically and your company information online?
  3. How strict will you want to go with customer information security?
  4. Will you be storing confidential data for your site members?
  5. Is there any requirement or consideration for security?
     For example, firewall limitation, necessity for using specific port.

b) c) Evidence of communication

We talked in our classroom about the schedule of meeting with the client and how to share our meeting agenda before the meeting.
We shared our agenda before the meeting via email, which is attached as the following.

The meeting was done via Zoom due to the COVID circumstance.
Attendants: Wei, Josephine, Lessley, Jina(me)
Date: 14 August 2020
Via: Zoom (online)

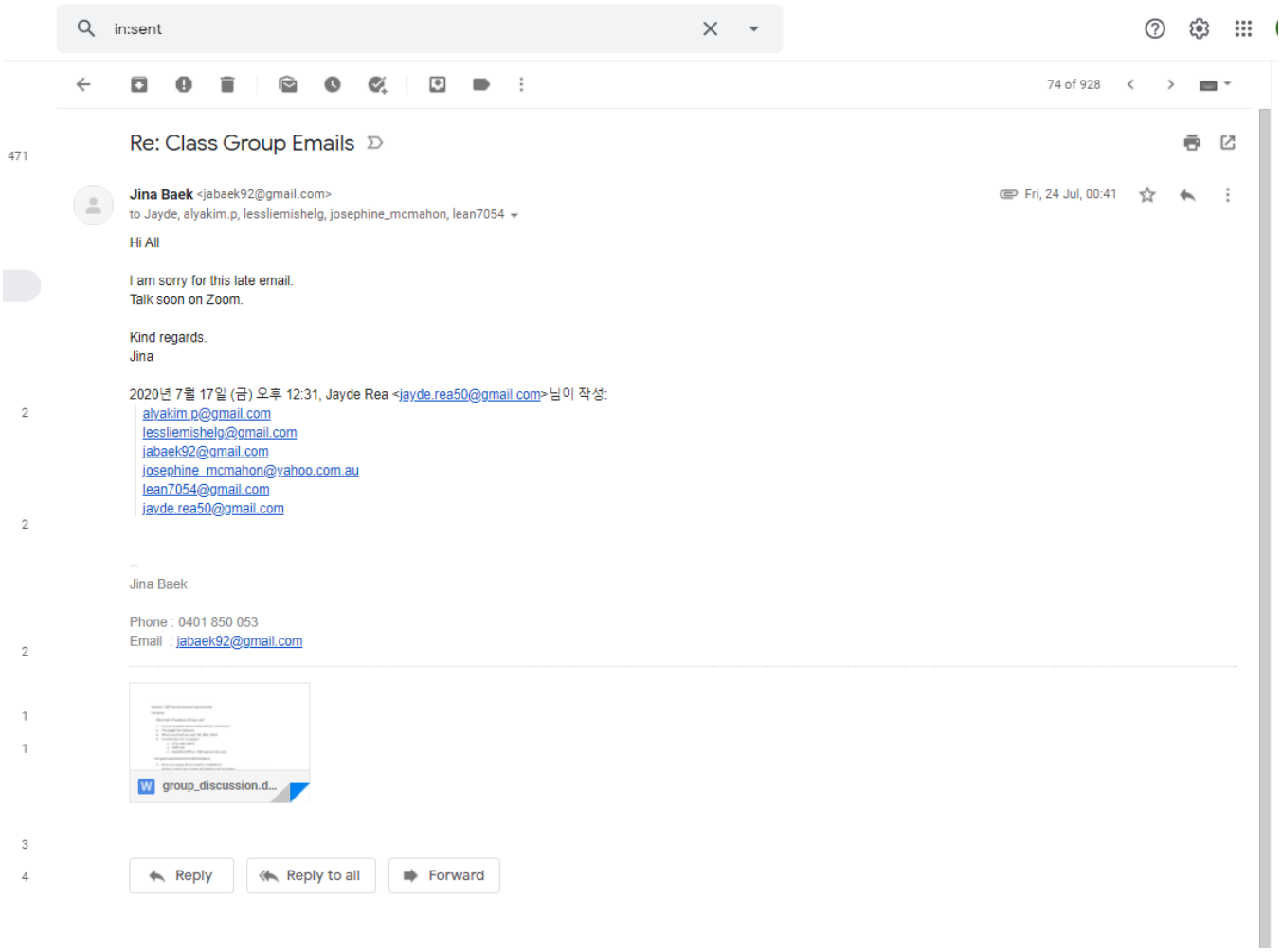


Figure 1 Proof of Sending Email

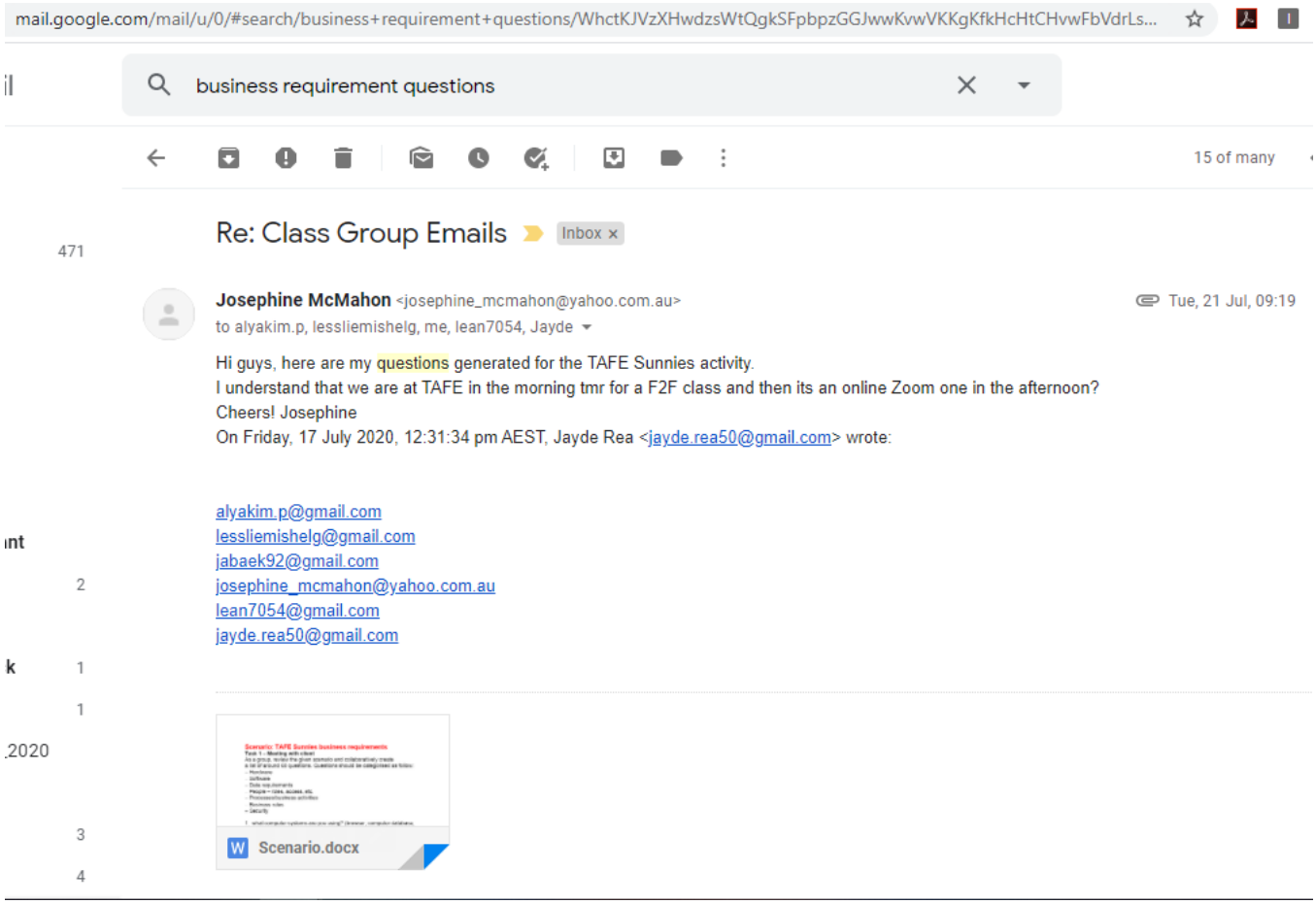


Figure 2 Proof of Sharing Agenda

d) Other four (4) methods to identify user needs

	Advantage	**Disadvantage**
Email Sharing	- A developer and a client do not need fix and share a specific time. They can use the time more efficiently. - Easy to share online materials such as document and image files. - Emails are written proofs as a common agreement	- Communication is asynchronous and need longer time to share all of the client's requirements than group meeting. - hard to catch emotional responses because people cannot see each other.
Online Zoom Video Meeting	- save travel time share requirements and computer screens for more information sharing. Everything can be recorded. - Communicate faster than writing methods such as email and chatting.	- The communication through the computer screen and microphone make difficult people understand the others' response, therefore it requires more effort to communicate efficiently.
Phone Interview	- Communicate faster than writing methods such as email and chatting.	- Multiple people cannot share the agenda

Online Group Chatting	- A developer and a client do not need fix and share a specific time. They can use the time more efficiently. - Easy to share online materials such as document and image files. - Chatting history is a written proof as a common agreement	- Everything is done through typing. Therefore, it is slower than talking. - hard to catch emotional responses because people cannot see each other.

2. Task 2 – Technical requirements

2.1 Hardware Requirements

For choosing appropriate hardware, the expected traffic amount, storage size and user number need to be considered. Also, hardware specification, such as CPU and RAM usage and hard disk performance, should be examined.

By the PPedology site, up to 7500 users means a big website, and this requires at least 200 GB disk, eight-core CPU and 16 GB RAM.

2.2 Software Requirements - DBMS Requirements

1. DBMS
   The client is considering MySQL server as a database management system and extra modules for security. They regarded Oracle but excluded due to the cost. MySQL Enterprise Edition is regarded.

2. Operating System
   CentOS is built on the source code of RedHat Enterprise Linux (REHL) and very popular for web service. This is stable, and a crash occurs rarely. It is also a free open source.

3. Webserver
   Apache webserver for Linux. This is a free open source version, and the client can reduce the building up cost.

2.3 Functional Requirement

1. Member registration

   A member is a customer registered as a TS member. They can track their shopping history and get an extra discount. For registration as a member, personal details and contact details are required. For example, name, phone number, address (Residential and Postal address), email address, login ID, password. The address consists of specific address including suburb, state, country and postcode. Date of birth will be required for particular countries to check the shopper age, which is over 18. This is related with business requirement and will be explained in details in next section, 2.4 Business

Requirement.

2. Login
    a. Admin login
       Admin users who manage products data can log in using login ID and password
    b. Member login
       Customer members who order sunglasses can log in using login ID and password

    The two types of authentication need to be provided on the website and the related information need to be stored in the database.

3. Product
    a. Register sunglasses information.
       EG. Sunglasses model name, mode no, material (crystal, metallic: titanium or other metallic, plastic), colour, style (sport, professional, special types), season (summer, winter), original price, discounted price, stock, comment, photos, provider.

    b. Modify information of sunglasses except for model no. If the admin user wants to change model no, they need to delete and insert it again.

    c. Delete a model.

4. Shopping cart.
    a. A user selects a model and sends it to the shopping cart to order.
    b. EG. User Login ID, model name, model no, the number of the item.
    c. When the purchase is decided and submitted, the payment screen will be displayed.
    d. Then the user can choose bank transfer or credit card payment,
    e. When the payment is completed, the receipt will be sent to the user email.
    f. The data saved in a shopping cart is temporary. Therefore, the data will be kept in the purchase history space.

5. Member purchase
   A member chooses items and adds them to the shopping cart. Then the member moves to the shopping cart and pays the total bill.

6. Guest purchase.
   A guest user is a customer who does not have a membership of TS. They can purchase TS items without registration. The same personal information and contact details as a member are required for the management of purchase and delivery. This information will be saved to a separated space (EG. database table).

    1) Guest user: They can purchase one time, and their information will be saved in a different place with the member users.

7. Provider
   This registers sunglasses provider info, such as name, contact info, contact person, address head office.

8. Purchase history.
   After a customer buys and pays an item, the purchase record is saved for tracking the purchase history and taxation purposes.

2.4 Business Requirement

1. Date of birth.
   For some specific country, purchasing products on the TS website is permitted to people over 18. When a customer inputs one of those countries, the date of birth field presents and is required to fill in. All other countries do not require a date of birth as input.

   This will be checked the date of birth for specific countries using a trigger. The specific countries should be registered in the database.

2. Member and Guest
   The customer can purchase a product as a member after registering or as a guest without a registration. The information about the two types of customers need to be stored and managed in the database.

3. Member Discount
   Customers are classified into two categories, a member and guest. Guests can buy a product at a full price, while members can get a 15% discount for every purchase.

   The programming needs to be implemented to put down the regular price by 15% discount.

4. Limitation of purchase number.
   A customer cannot buy more than a given number of one model. This number is set to 3 for implementation. This will be checked in a trigger and the number will be specified in it.

5. The data saved in a shopping cart is temporary. When the purchase is completed, the purchase record will be moved to a permanent storage such as the purchase history table. This is for tracking the purchase history and taxation purposes.

   The programming application will be implemented for this.

   Example(Pseudo code):

   ```
   begin transaction
   SELECT data FROM ShoppingCart table
   INSERT  them INTO Order(history) table
   DELETE them FROM shoppingCart table
   end transaction
   ```

6. When the payment is completed, the receipt will be sent to the user email.
   This will be implemented by programming.

2.5 Security Requirement

. The password for authentication needs to be encrypted and saved in the database. and the session management to access to the personal information securely needs to be implemented.

3. Task 3 – Conceptual model

1. **Customer:** all customers table. This include all members registered as a member of TAFE Sunnies (TS), all guests who do not register but purchase a product and administrators

   customerId: PK
   memberFlag
   firstName
   lastName
   phoneNo
   email
   address
   postId: FK
   dateOfBirth: only necessary for some country
   loginId
   password: encrypted password
   adminFlag: [Yes: administrator, No: regular member]

2. **Post:** all surburbs and post code information

   postId: PK
   postcode: real postcode
   suburb
   city
   state
   country

3. **ShoppingCart**: customer's shopping cart data, temporary data.
   After checking that the payment is done, the data is saved to Order, OrderDetail, Payment table for permanent recording. Then this data will be deleted from the ShoppingCart table.

   shopCartId: PK
   productId: FK
   quantity

4. **Order**: permanent order data table, all paid purchases are saved as permanent records.

   orderId: PK
   customerId: FK [memberId of Member table, guestId of Guest table]
   memberflag: [member, Guest]
   orderDate
   shippedDate

paymentFlag: [paid, waiting for transferring]
paymentType [bank transfer, card]
cardNo
expireDate: card expire date
comment

5. **OrderDetail**: order details table, products and quantitys per order

   orderDetailId: PK
   prductId: FK
   orderId: FK
   quantity
   paidPrice

6. **Product**: all sunglasses products that TS sells

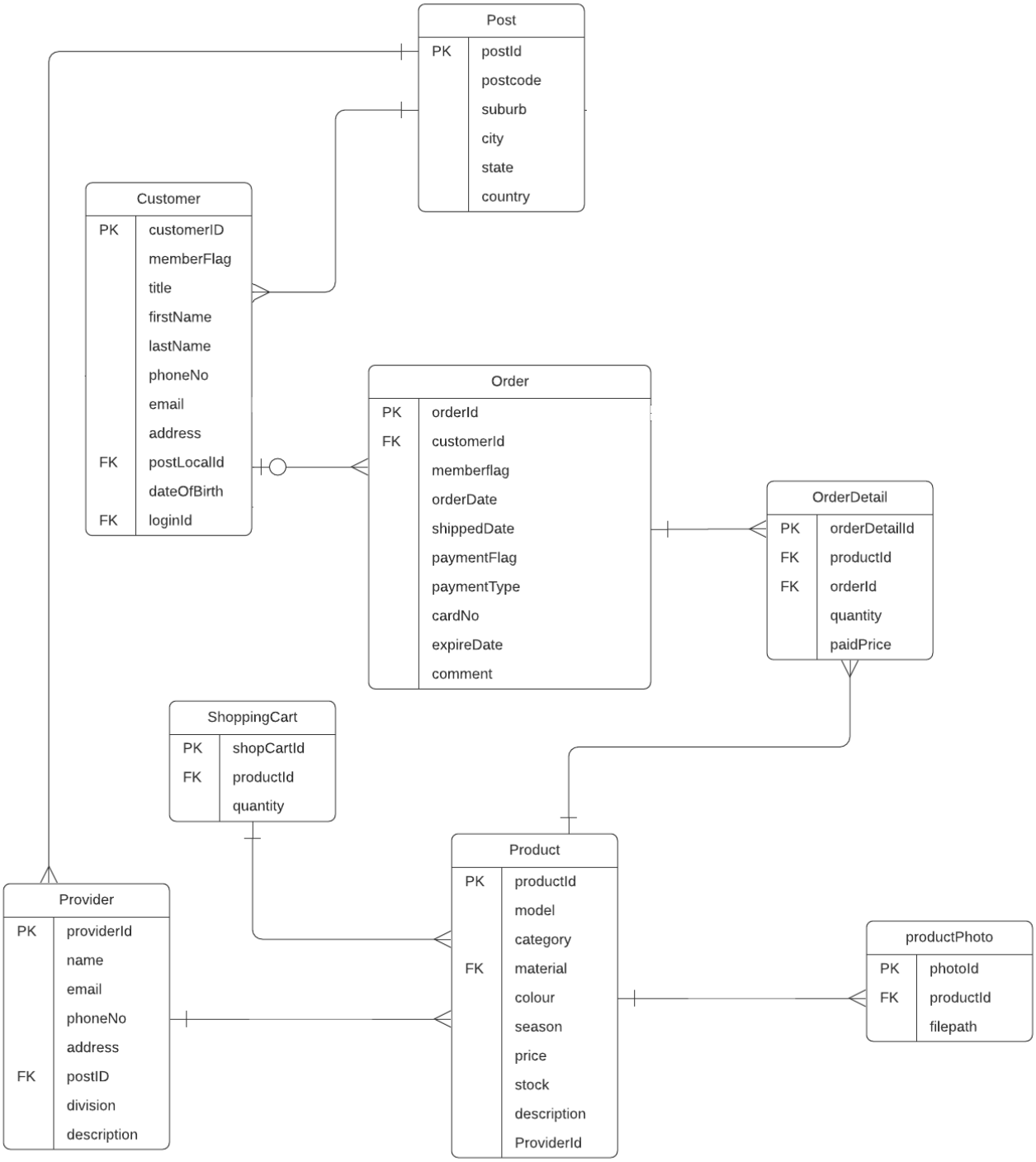
   productId: PK
   model
   category: [sport, professional, special types]
   material
   colour
   season:[summer, winter]
   price
   stock
   description
   providerId: FK

7. **productPhoto**: photos to a specific sunglasses.

   photoId: PK
   productId: FK
   filepath

8. **Provider**: table of providers who manufacture and provide sunglasses to TS.

   providerId: PK
   name
   email
   phoneNo
   address
   postId: FK
   division [branch, headoffice]
   description



Client Feedback

Date: 11/09/2020
Via: face to face meeting
Content:
The client commented many data are duplicated specifically Post and Product tables.
The authentication system is not provided. In other words, the login function need to be implemented and the different level of user management including administration users and general customers should be contained.

4. Task 4 – Logical data model

4.1 Normalisation process

1NF: Primary key exists, and all columns are atomic
2NF: A table is in 1NF.
All columns are fully dependent on the primary key.
3NF: A table is in 2NF. All other columns except the primary key are non-transitively dependent on the primary key.

4.1.1 Customer

The Customer table has the primary key, and all columns are atomic and fully dependent on the primary key. However, loginID, password is transitively dependent and adminFlag repeats unnecessary. Member needs to be splited to three table, Member, Login and Admin tables.

table-1) Customer

customerId: int(10) PK
memberFlag: varchar(10)
title: varchar(4) [Mr, Mrs, Ms, Miss]
firstName: varchar(20)
lastName: varchar(20)
phoneNo: varchar(12)
email: varchar(50)
address: varchar(30)
postLocalId: int(10) FK
dateOfBirth:date (only necessary for some country)
loginId: varchar(15)

satisfies 3NF.

- For some countries, people over 18 years old can purchase online. dateOfBirth column is set for those countries. The countries can be found in PostLocal table using localId. When member data is registered, a php program will check the member's country and ask of users to input their date of birth.

table-2) Login: login ID and password of members and administrators

loginID: varchar(15) PK
password: varchar(50) encrypted password

satisfies 3NF.

table-3) Admin: administrator information

adminId: int(10) PK
name: varchar(30)
loginId: varchar(15) FK

satisfies 3NF.

- An administrator registered in Admin table can add a new product, modify its details such as price, and delete existing products. These jobs can be done only through the web interface.

4.1.2 Post

The Post table has the primary key, and all columns are atomic and fully dependent on the primary key. However, state and country columns are transitively dependent. To reduce redundancy of state and country data, this is split into **PostLocal** and **PostGlobal**

table-1) PostLocal: surburbs, city and post code in states or cities.

postLocalId: int(10) PK
postcode: varchar(10)
suburb: varchar(20)
postGlobalId: int(5) FK

satisfies 3NF.

table-2) PostGlobal: city, state and country name for federated countries.
satisfies 3NF.

postGlobalId: int(5) PK
city: varchar(15)
state: varchar(10)
country: varchar(15)

satisfies 3NF.

4.1.3 ShoppingCart

shopCartId: int(5) PK
productId: int(10) FK
quantity: int(5)

satisfies 3NF.

- A customer cannot buy more than a given number of one model.
  If the quantity of a shopping cart item is more than 3 (not fixed yet), an error will display.
  This can be implemented by a trigger. When a record is inserted into the shopping cart table, trigger checks the quantity and set an error message.

- After the payment is committed, the shopping cart data will insert the data into Order, OrderDetails for permanent history and delete the shopping cart data. This will be done by application (This work can be done in php applications.)

4.1.4 Order

The Order table has the primary key, and all columns are atomic and fully dependent on the primary key. However, paymentFlag, cardNo is transitively dependent to orderID. Therefore, Order is split into Order, paymentCash and PaymentCard tables.

table-1) Order

orderId: int(20) - PK
customerId: int(10) - FK
memberflag: varchar(10) [member, Guest]
orderDate: date
shippedDate:date
paymentId: int(10) FK
comment varchar(500)

satisfies 3NF.

table-2) Payment: payment information

paymentId: int(10) PK
paymentType: varchar(10)
paidDate: date
cardInfoID: int(10) unsigned

satisfies 3NF.

table-3) Card card information

paymentId: int(10) unsigned PK
cardtype: varchar(10)
cardNo: int(16)
expireDate: date

satisfies 3NF.

4.1.5 OrderDetail

orderDetailId: int(20) PK
prductId: varchar(20) FK
orderId: int(20): FK
quantity: int(5)
paidPrice: decimal(13, 4)

satisfies 3NF.

- When a member purchases products, 15% discounted amount of the price in the Product table is set to the paidPrice column. If guest purchases, the price of product table is set to the paidPrice column. This will be done by programming

4.1.6 Product

The Product table has the primary key, and all columns are atomic and fully dependent on the primary key. However, model, description, provider, category, material, colour and season are transitively dependent to productId, the Product table is split into Product, Model, Category, Material tables.

table-1) Product

productId: int(10) PK
modelId: int(5) FK
categoryId: int(4) FK
materialId: int(4) FK
price: decimal(13, 4)
stock: int(5)

satisfies 3NF.

table-2) Model

modelId: int(5) PK
model: varchar(20)
providerId: int(10) FK
description: varchar(500)

satisfies 3NF.

table-3) Category

categoryId: int(4) PK
category: varchar(10) [sport, professional, special types]
season: varchar(10) [summer, winter]

satisfies 3NF.

table-4) Material

materialId: int(4) PK
material: varchar(10) [crystal, metallic, titanium, plastic]
colour: varchar(10)

satisfies 3NF.

4.1.7 productPhoto

photos to a specific sunglasses

photoId: int(15) PK
productId: int(10) FK
filepath: varchar(100)

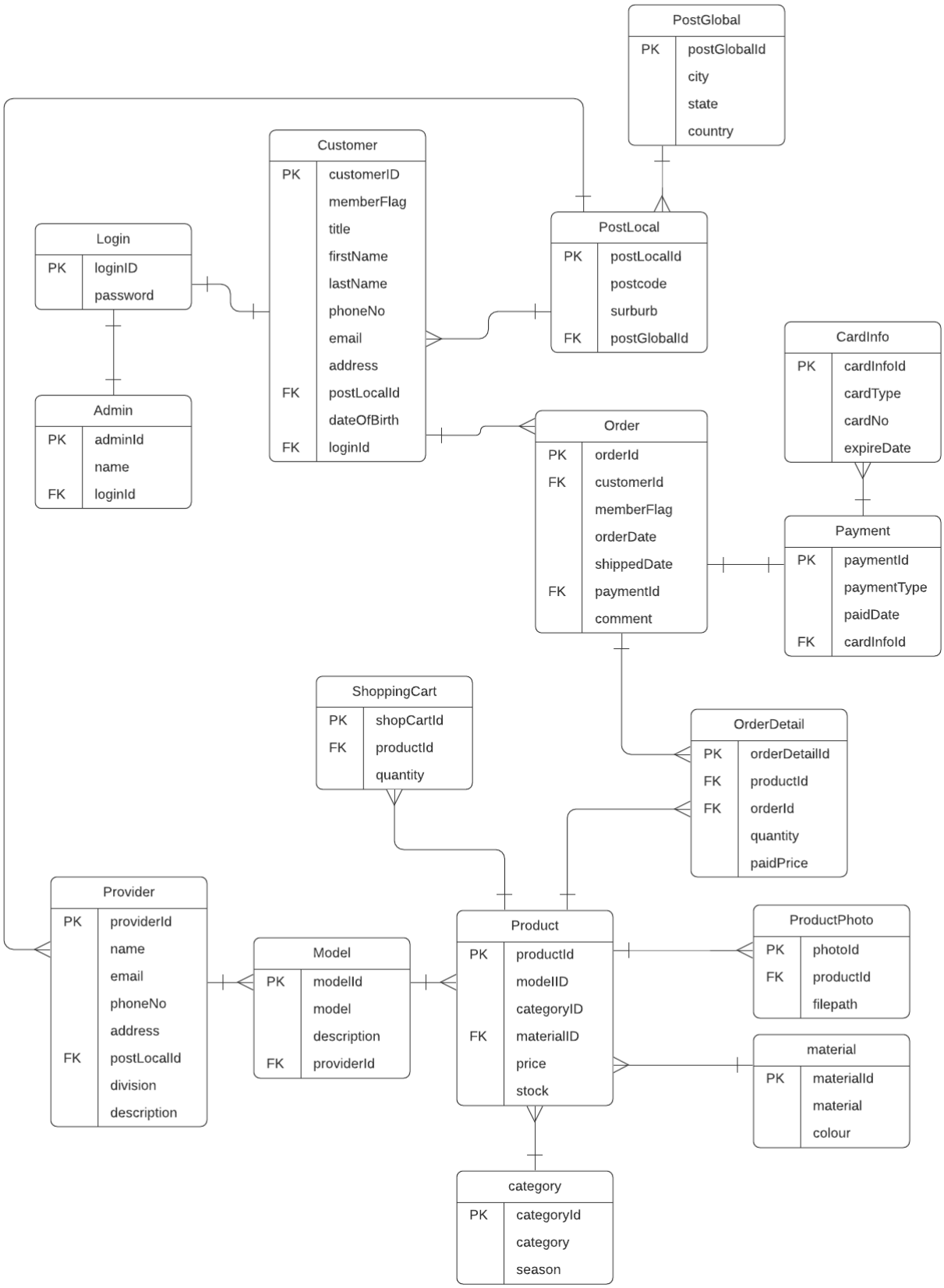
satisfies 3NF.

4.1.8 Provider

providerId: int(10) PK
name: varchar(30)
email: varchar(50)
phoneNo: varchar(12)
address: varchar(30)
postLocalId: int(10) FK
division: varchar(10) [branch, headoffice]
description: varchar(500)

satisfies 3NF.

4.2 Final ERD


PostGlobal
PK postGlobalId
city
state
country
Customer
PK customerID
memberFlag
title
firstName
lastName
phoneNo
email
address
FK postLocalId
dateOfBirth
FK loginId
Login
PK loginID
password
PostLocal
PK postLocalId
postcode
surburb
FK postGlobalId
CardInfo
PK cardInfoId
cardType
cardNo
expireDate
Admin
PK adminId
name
FK loginId
Order
PK orderId
FK customerId
memberFlag
orderDate
shippedDate
FK paymentId
comment
Payment
PK paymentId
paymentType
paidDate
FK cardInfoId
ShoppingCart
PK shopCartId
FK productId
quantity
OrderDetail
PK orderDetailId
FK productId
FK orderId
quantity
paidPrice
Provider
PK providerId
name
email
phoneNo
address
FK postLocalId
division
description
Model
PK modelId
model
description
FK providerId
Product
PK productId
modelID
categoryID
FK materialID
price
stock
ProductPhoto
PK photoId
FK productId
filepath
material
PK materialId
material
colour
category
PK categoryId
category
season

4.3 Data Dictionary

admin

Column	Type	Null	Default	Comments
adminId *(Primary)*	int(10)	No		
name	varchar(30)	No		
loginId	varchar(15)	Yes	*NULL*	

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	adminId	2	A	No	
fk_loginid_admin	BTREE	No	No	loginId	2	A	Yes	

cardinfo

Column	Type	Null	Default	Comments
cardInfoId *(Primary)*	int(10)	No		
cardtype	varchar(10)	No		
cardNo	varchar(16)	Yes	*NULL*	
expireDate	date	No		

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	cardInfoId	0	A	No	
cardNo	BTREE	Yes	No	cardNo	0	A	Yes	

category

Column	Type	Null	Default	Comments
categoryId *(Primary)*	int(4)	No		
category	varchar(10)	No		
season	varchar(10)	No		

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	categoryId	2	A	No	

checkagecountries

Column	Type	Null	Default	Comments
countryId *(Primary)*	int(5)	No		
name	varchar(30)	Yes	*NULL*	

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	countryId	2	A	No	

customer

Column	Type	Null	Default	Comments
customerId *(Primary)*	int(10)	No		
memberFlag	varchar(10)	No		

title	varchar(4)	Yes	*NULL*	
firstName	varchar(20)	No		
lastName	varchar(20)	No		
phoneNo	varchar(12)	No		
email	varchar(50)	No		
address	varchar(30)	No		
postLocalId	int(10)	Yes	*NULL*	
dateOfBirth	date	Yes	*NULL*	
loginId	varchar(15)	Yes	*NULL*	

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	customerId	2	A	No	
fk_postlocalid_customer	BTREE	No	No	postLocalId	1	A	Yes	
fk_loginid_customer	BTREE	No	No	loginId	2	A	Yes	

login

Column	Type	Null	Default	Comments
loginId *(Primary)*	varchar(15)	No		
password	varchar(50)	No		

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	loginId	0	A	No	

material

Column	Type	Null	Default	Comments
materialId *(Primary)*	int(4)	No		
material	varchar(10)	No		
colour	varchar(10)	No		

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	materialId	4	A	No	

model

Column	Type	Null	Default	Comments
modelId *(Primary)*	int(5)	No		
model	varchar(20)	No		
description	varchar(500)	Yes	*NULL*	
providerId	int(10)	Yes	*NULL*	

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	modelId	2	A	No	
fk_providerid_model	BTREE	No	No	providerId	1	A	Yes	

ordercart

Column	Type	Null	Default	Comments
orderCartID *(Primary)*	int(10)	No		
orderId	int(10)	Yes	*NULL*	
shopCartId	int(5)	Yes	*NULL*	

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	orderCartID	0	A	No	
fk_orderId_ordercart	BTREE	No	No	orderId	0	A	Yes	
fk_shopCartId_ordercart	BTREE	No	No	shopCartId	0	A	Yes	

orderdetail

Column	Type	Null	Default	Comments
orderDetailId *(Primary)*	int(10)	No		
productId	int(10)	Yes	*NULL*	
orderId	int(10)	Yes	*NULL*	
quantity	int(5)	No		
paidPrice	decimal(13,4)	Yes	*NULL*	

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	orderDetailId	2	A	No	
fk_productid_orderdetail	BTREE	No	No	productId	2	A	Yes	
fk_orderid_orderdetail	BTREE	No	No	orderId	1	A	Yes	

orders

Column	Type	Null	Default	Comments
orderId *(Primary)*	int(10)	No		
customerId	int(10)	Yes	*NULL*	
memberFlag	varchar(10)	No		
orderDate	date	No		
shippedDate	date	Yes	*NULL*	
paymentId	int(10)	Yes	*NULL*	
comment	varchar(500)	Yes	*NULL*	

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	orderId	2	A	No	
fk_customerid_order	BTREE	No	No	customerId	2	A	Yes	

payment

Column	Type	Null	Default	Comments
paymentId *(Primary)*	int(10)	No		
paymentType	varchar(10)	Yes	*NULL*	
paidDate	date	Yes	*NULL*	
cardInfoId	int(10)	Yes	*NULL*	

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	paymentId	2	A	No	
fk_cardinfoid_payment	BTREE	No	No	cardInfoId	2	A	Yes	

postglobal

Column	Type	Null	Default	Comments
postGlobalId *(Primary)*	int(5)	No		
city	varchar(20)	Yes	*NULL*	
state	varchar(20)	Yes	*NULL*	
country	varchar(20)	Yes	*NULL*	

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	postGlobalId	0	A	No	

postlocal

Column	Type	Null	Default	Comments
postLocalId *(Primary)*	int(10)	No		
postcode	varchar(10)	No		
suburb	varchar(20)	No		
postGlobalId	int(5)	Yes	*NULL*	

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	postLocalId	0	A	No	
fk_globalId_postlocal	BTREE	No	No	postGlobalId	0	A	Yes	

product

Column	Type	Null	Default	Comments
productId *(Primary)*	int(10)	No		
modelId	int(5)	Yes	*NULL*	
categoryId	int(4)	Yes	*NULL*	
materialId	int(4)	Yes	*NULL*	
price	decimal(13,4)	Yes	*NULL*	
stock	int(5)	No	0	

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	productId	2	A	No	
fk_modelid_product	BTREE	No	No	modelId	1	A	Yes	
fk_categoryid_product	BTREE	No	No	categoryId	1	A	Yes	
fk_materialid_product	BTREE	No	No	materialId	2	A	Yes	

productphoto

Column	Type	Null	Default	Comments
photoId *(Primary)*	int(11)	No		
productId	int(10)	Yes	*NULL*	

filepath	varchar(100)	No		

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	photoId	2	A	No	
fk_productid_productphoto	BTREE	No	No	productId	2	A	Yes	

provider

Column	Type	Null	Default	Comments
providerId *(Primary)*	int(10)	No		
name	varchar(30)	No		
email	varchar(50)	No		
phoneNo	varchar(12)	No		
address	varchar(30)	No		
postLocalId	int(10)	Yes	*NULL*	
division	varchar(20)	Yes	*NULL*	
description	varchar(500)	Yes	*NULL*	

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	providerId	2	A	No	
fk_postlocalid_provider	BTREE	No	No	postLocalId	1	A	Yes	

shoppingcart

Column	Type	Null	Default	Comments
shopCartId *(Primary)*	int(5)	No		
productID	int(10)	Yes	*NULL*	
quantity	int(5)	Yes	*NULL*	

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	shopCartId	2	A	No	
fk_productid_shoppingcart	BTREE	No	No	productID	2	A	Yes	

4.4 Approval to the client

Final version of ERD database

Manuela Perez

Final version of ERD database

Dear Mrs Manuela Perez

I ask you to check the final version of ERD for your database implementation.
All tables are changed to 3rd Normal Forms after normailisation process.
I applied your functional and business requirements completely.
However, if you find something insufficient or necessity of modification, please do not hesitate to contact me.

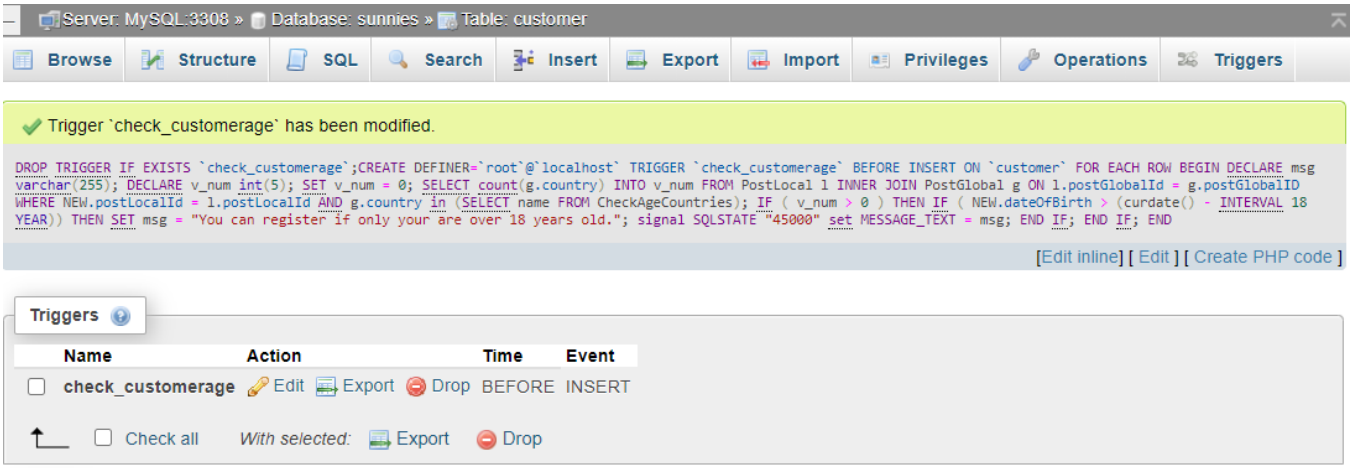
Thank you in advance.
Regards

Jina Baek
Database developer
phone: 0404 111 222
email: jinaemail@email.com

5. Task 5 – Business rules and constraints

5.1 Business rules

1. Date of birth.
   For some specific country, purchasing products on the TS website is permitted to people over 18. When a customer inputs one of those countries, the date of birth field presents and is required to fill in. All other countries do not require a date of birth as input.



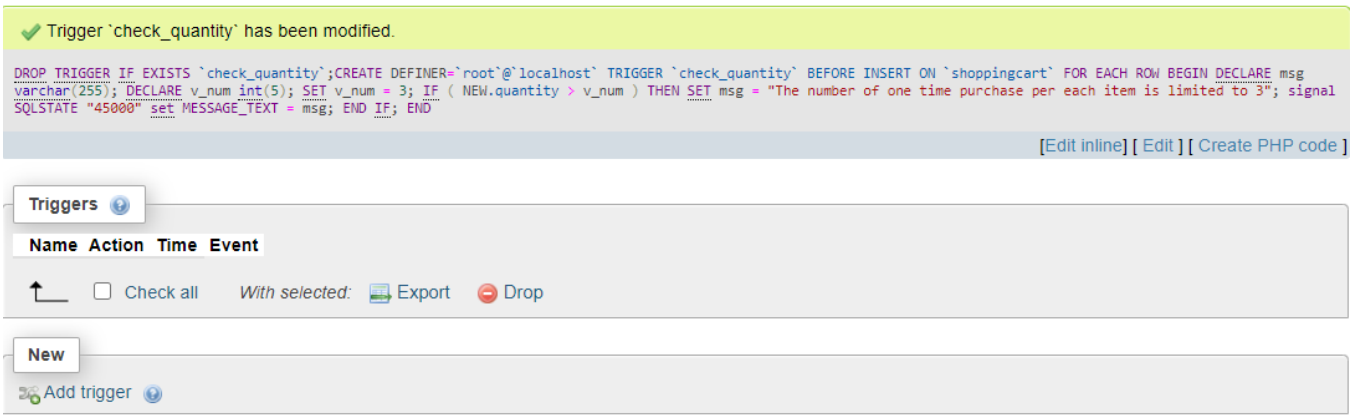
2. Member and Guest
   The customer can purchase a product as a member after registering or as a guest without a registration. The information about the two types of customers need to be stored and managed in the database.

This is implemented as the MemberFlag column in the Customer table.

3. Member Discount
   Customers are classified into two categories, a member and guest. Guests can buy a product at a full price, while members can get a 15% discount for every purchase.

   The programming needs to be implemented to put down the regular price by 15% discount.

4. Limitation of purchase number.
   A customer cannot buy more than a given number of one model. This number is set to 3 for implementation. This will be checked in a trigger and the number will be specified in it.



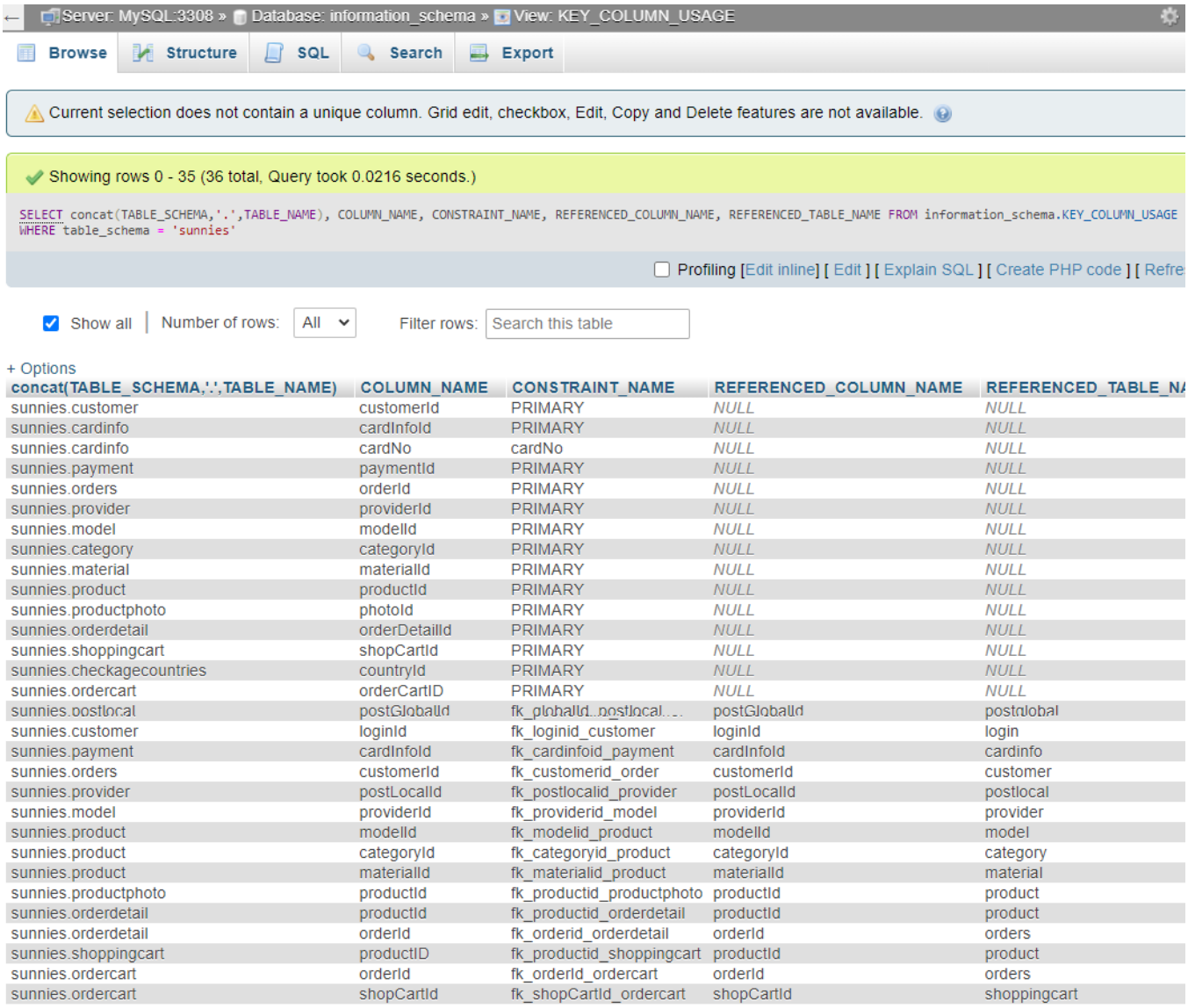
5. The data saved in a shopping cart is temporary. When the purchase is completed, the purchase record will be moved to a permanent storage such as the purchase history table. This is for tracking the purchase history and taxation purposes.

   The programming application will be implemented for this.

   Example(Pseudo code):
   ```
   begin transaction
   SELECT data FROM ShoppingCart table
   INSERT  them INTO Order(history) table
   DELETE them FROM shoppingCart table
   end transaction
   ```

6. When the payment is completed, the receipt will be sent to the user email. This also will be implemented by programming.

5.2 Integrity constraints and Referential integrity constraints



concat(TABLE_SCHEMA,'.',TABLE_NAME)	COLUMN_NAME	CONSTRAINT_NAME	REFERENCED_COLUMN_NAME	REFERENCED_TABLE_NA
sunnies.customer	customerId	PRIMARY	*NULL*	*NULL*
sunnies.cardinfo	cardInfoId	PRIMARY	*NULL*	*NULL*
sunnies.cardinfo	cardNo	cardNo	*NULL*	*NULL*
sunnies.payment	paymentId	PRIMARY	*NULL*	*NULL*
sunnies.orders	orderId	PRIMARY	*NULL*	*NULL*
sunnies.provider	providerId	PRIMARY	*NULL*	*NULL*
sunnies.model	modelId	PRIMARY	*NULL*	*NULL*
sunnies.category	categoryId	PRIMARY	*NULL*	*NULL*
sunnies.material	materialId	PRIMARY	*NULL*	*NULL*
sunnies.product	productId	PRIMARY	*NULL*	*NULL*
sunnies.productphoto	photoId	PRIMARY	*NULL*	*NULL*
sunnies.orderdetail	orderDetailId	PRIMARY	*NULL*	*NULL*
sunnies.shoppingcart	shopCartId	PRIMARY	*NULL*	*NULL*
sunnies.checkagecountries	countryId	PRIMARY	*NULL*	*NULL*
sunnies.ordercart	orderCartID	PRIMARY	*NULL*	*NULL*
sunnies.postlocal	postGlobalId	fk_globalId_postlocal...	postGlobalId	postglobal
sunnies.customer	loginId	fk_loginid_customer	loginId	login
sunnies.payment	cardInfoId	fk_cardinfoid_payment	cardInfoId	cardinfo
sunnies.orders	customerId	fk_customerid_order	customerId	customer
sunnies.provider	postLocalId	fk_postlocalid_provider	postLocalId	postlocal
sunnies.model	providerId	fk_providerid_model	providerId	provider
sunnies.product	modelId	fk_modelid_product	modelId	model
sunnies.product	categoryId	fk_categoryid_product	categoryId	category
sunnies.product	materialId	fk_materialid_product	materialId	material
sunnies.productphoto	productId	fk_productid_productphoto	productId	product
sunnies.orderdetail	productId	fk_productid_orderdetail	productId	product
sunnies.orderdetail	orderId	fk_orderid_orderdetail	orderId	orders
sunnies.shoppingcart	productID	fk_productid_shoppingcart	productId	product
sunnies.ordercart	orderId	fk_orderId_ordercart	orderId	orders
sunnies.ordercart	shopCartId	fk_shopCartId_ordercart	shopCartId	shoppingcart

All foreign keys are implemented in ON UPDATE CASCADE ON DELETE SET NULL.
All NOT NULL Constraint are specified in the data dictionary.

5.3 Semantic and other constraints

The TRIGGER check_customerage checks the business rule that limits the purchase of people who are under 18 years old.
The TRIGGER check_number checks the purchase limitation number per each item.

Other business constraints such as the limitation of purchase number will be applied by programming.

5.4 Indexes

Showing rows 0 - 24 (33 total, Query took 0.0011 seconds.)

```
SELECT table_name, column_name, index_name, index_type FROM information_schema.STATISTICS WHERE table_schema = 'sunnies'
```

Profiling [Edit inline] [Edit] [Explain S

1 > >> | Show all | Number of rows: 25 Filter rows: Search this table

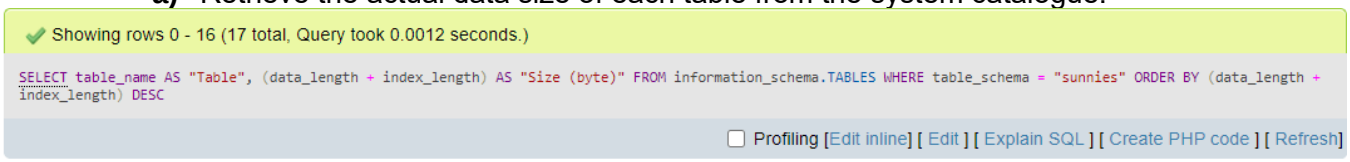
+ Options

TABLE_NAME	COLUMN_NAME	INDEX_NAME	INDEX_TYPE
admin	loginId	fk_loginid_admin	BTREE
admin	adminId	PRIMARY	BTREE
cardinfo	cardNo	cardNo	BTREE
cardinfo	cardInfoId	PRIMARY	BTREE
category	categoryId	PRIMARY	BTREE
checkagecountries	countryId	PRIMARY	BTREE
customer	loginId	fk_loginid_customer	BTREE
customer	postLocalId	fk_postlocalid_customer	BTREE
customer	customerId	PRIMARY	BTREE
login	loginId	PRIMARY	BTREE
material	materialId	PRIMARY	BTREE
model	providerId	fk_providerid_model	BTREE
model	modelId	PRIMARY	BTREE
orderdetail	orderId	fk_orderid_orderdetail	BTREE
orderdetail	productId	fk_productid_orderdetail	BTREE
orderdetail	orderDetailId	PRIMARY	BTREE
orders	customerId	fk_customerid_order	BTREE
orders	orderId	PRIMARY	BTREE
payment	cardInfoId	fk_cardinfoid_payment	BTREE
payment	paymentId	PRIMARY	BTREE
postglobal	postGlobalId	PRIMARY	BTREE
postlocal	postGlobalId	fk_globalId_postlocal	BTREE
postlocal	postLocalId	PRIMARY	BTREE
product	categoryId	fk_categoryid_product	BTREE
product	materialId	fk_materialid_product	BTREE

1 > >> | Show all | Number of rows: 25 Filter rows: Search this table

5.5 Estimation of the approximate size of the completed database

a) Retrieve the actual data size of each table from the system catalogue.



Show all | Number of rows: 25 Filter rows: Search this table

+ Options

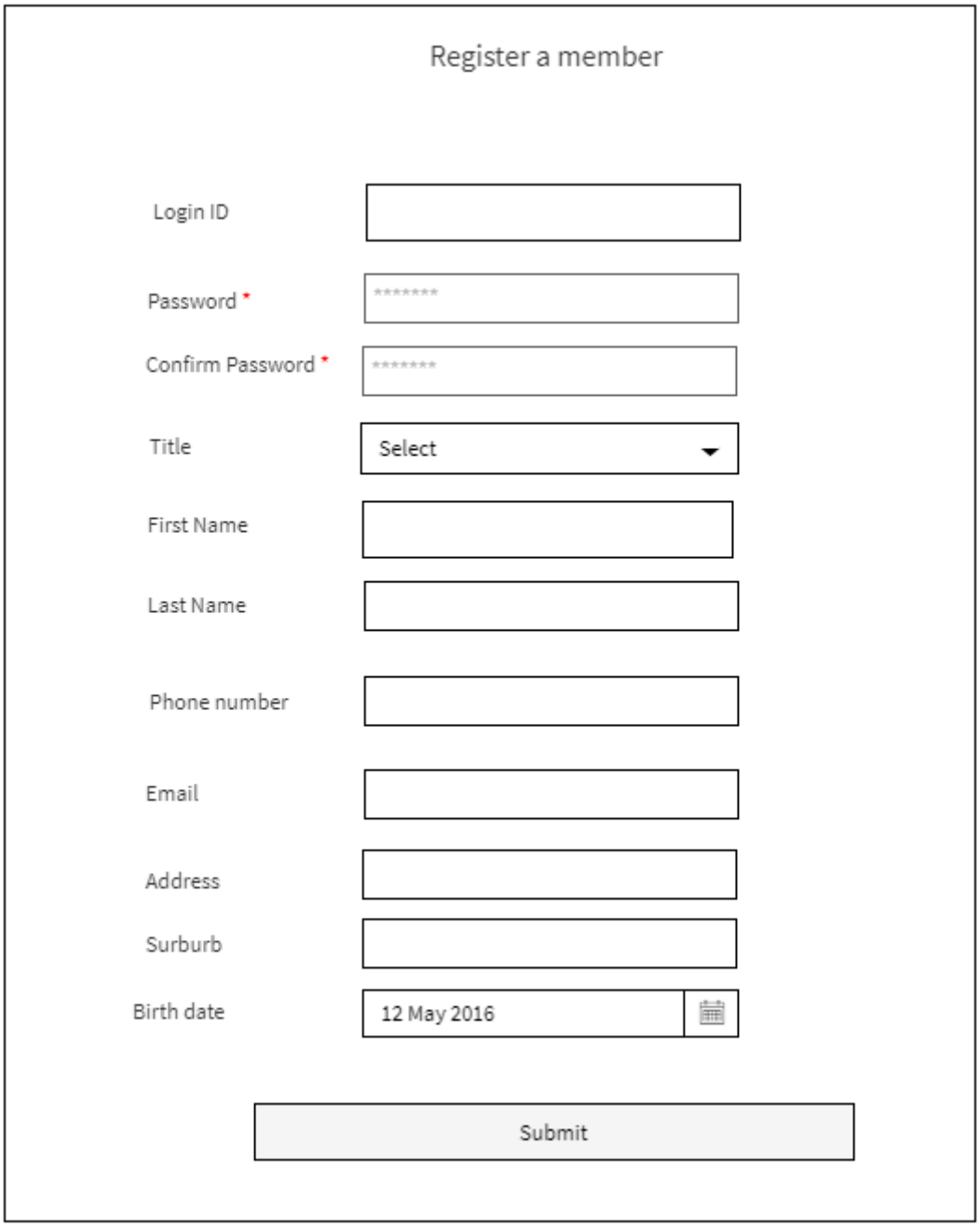
Table	Size (byte)
product	65536
customer	49152
orderdetail	49152
admin	32768
cardinfo	32768
model	32768
orders	32768
payment	32768
postlocal	32768
productphoto	32768
provider	32768
shoppingcart	32768
category	16384
checkagecountries	16384
login	16384
material	16384
postglobal	16384

b) Find one row size through dividing the size by the number of records
c) Multiply one row size with 5000(estimated record numbers)
d) The whole estimated size is **402Mbytes**

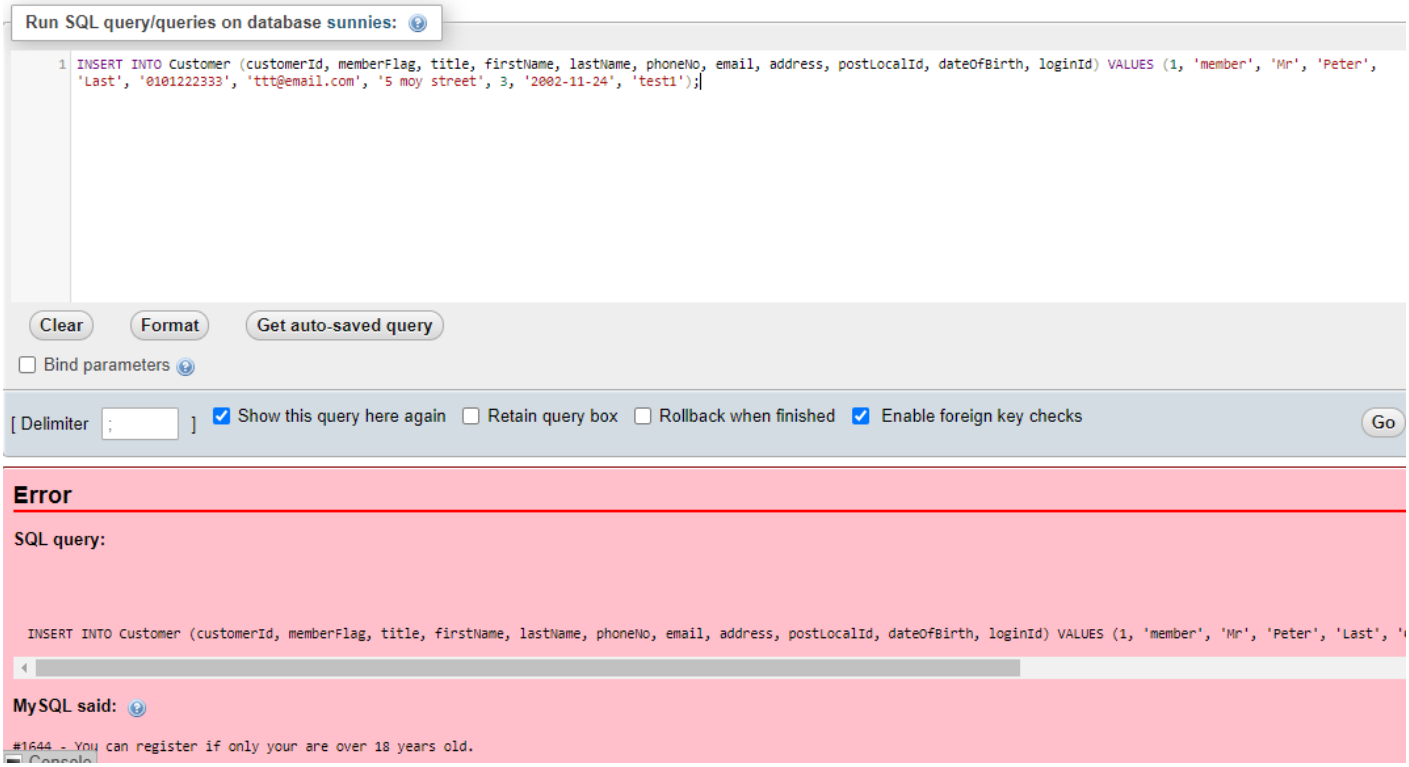
Table	Size(Byte)	count(*)	one row size	5000 rows size(byte)	5000 row (MB)
product	65536	5	13107.2	65536000	31.62055336
customer	49152	4	12288	61440000	29.64426877
orderdetail	49152	6	8192	40960000	19.76284585
admin	32768	2	16384	81920000	39.5256917
cardinfo	32768	3	10922.66667	54613333.33	26.35046113
model	32768	3	10922.66667	54613333.33	26.35046113
orders	32768	4	8192	40960000	19.76284585
payment	32768	4	8192	40960000	19.76284585
postlocal	32768	4	8192	40960000	19.76284585
productphoto	32768	3	10922.66667	54613333.33	26.35046113
provider	32768	2	16384	81920000	39.5256917
shoppingcart	32768	2	16384	81920000	39.5256917
category	16384	3	5461.333333	27306666.67	13.17523057
checkagecountries	16384	2	8192	40960000	19.76284585
login	16384	5	3276.8	16384000	7.90513834
material	16384	4	4096	20480000	9.881422925
postglobal	16384	3	5461.333333	27306666.67	13.17523057
					401.8445323

6. User interface

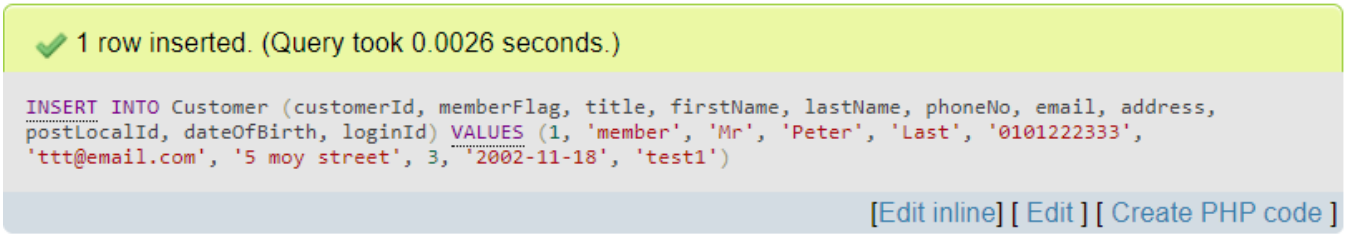
6.1 Member registration



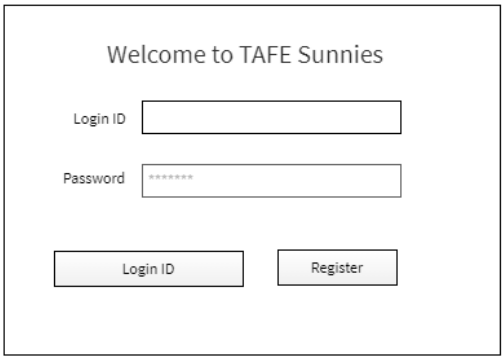
For a specific country, when registration of a person who is under 18 years is prohibited by a trigger. The countries are registered in the CheckAgeCountries table.



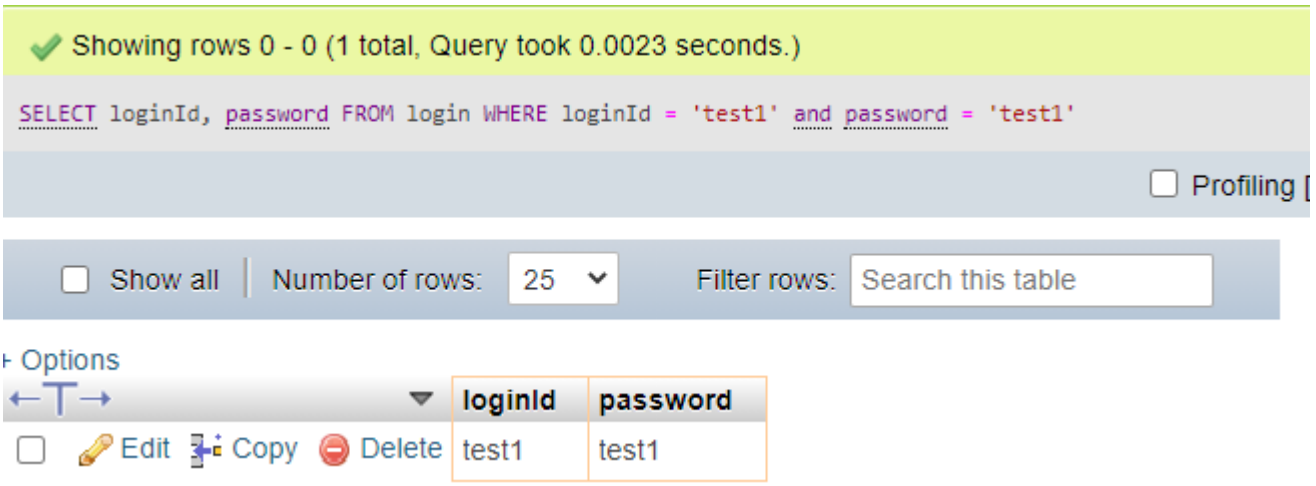
After changing the date of birth to 2002-11-18, it is inserted successfully.



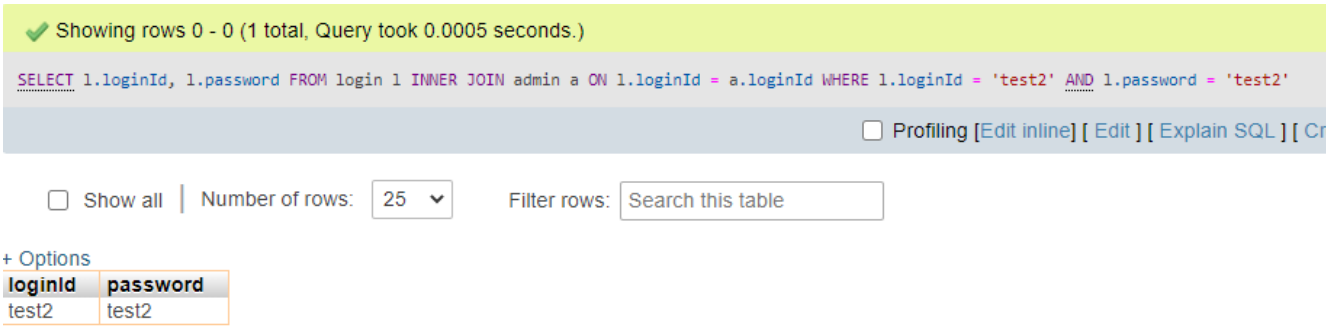
6.2 Login



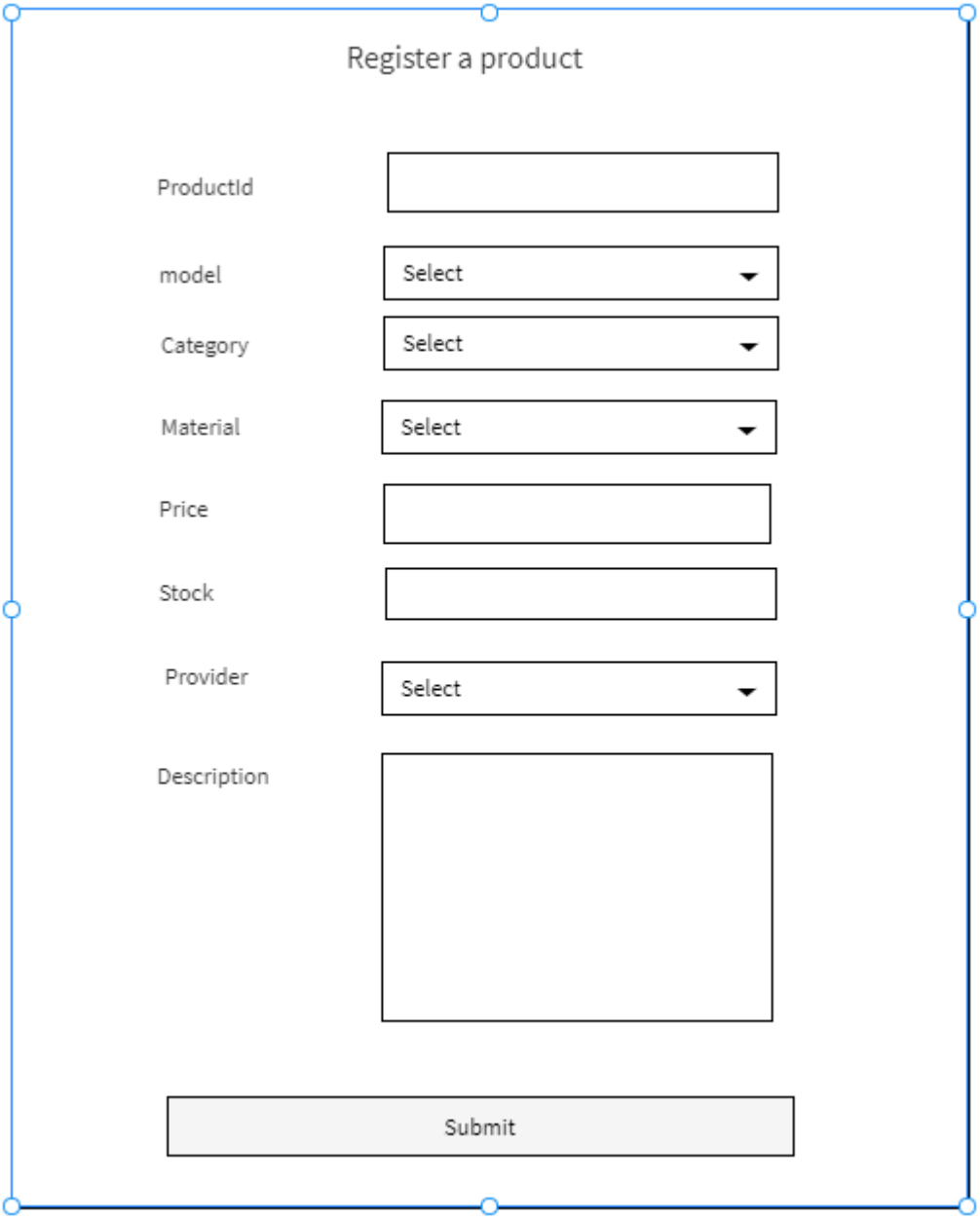
Member login check



Administrator login check



6.3 Product Registration

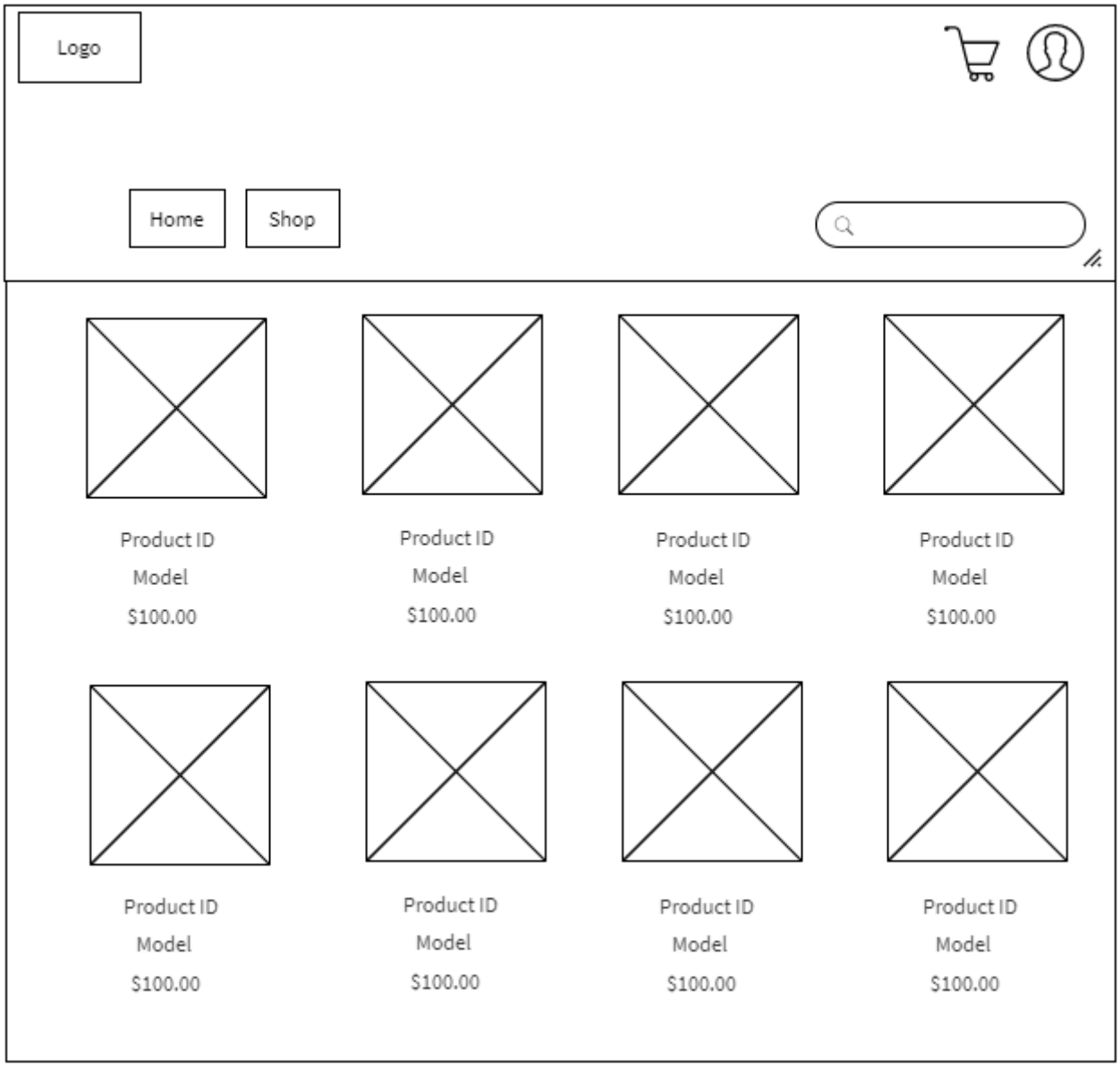


1 row inserted. (Query took 0.0025 seconds.)

```
INSERT INTO Product (productId, modelId, categoryId, materialId, price, stock) VALUES (4, 2, 2, 22,
250.00, 21)
```

[Edit inline] [Edit] [Create PHP code]

6.4 Product List



Showing rows 0 - 3 (4 total, Query took 0.0006 seconds.)

```
SELECT productId, model, price FROM Product p INNER JOIN Model m ON p.modelId = m.modelId
```

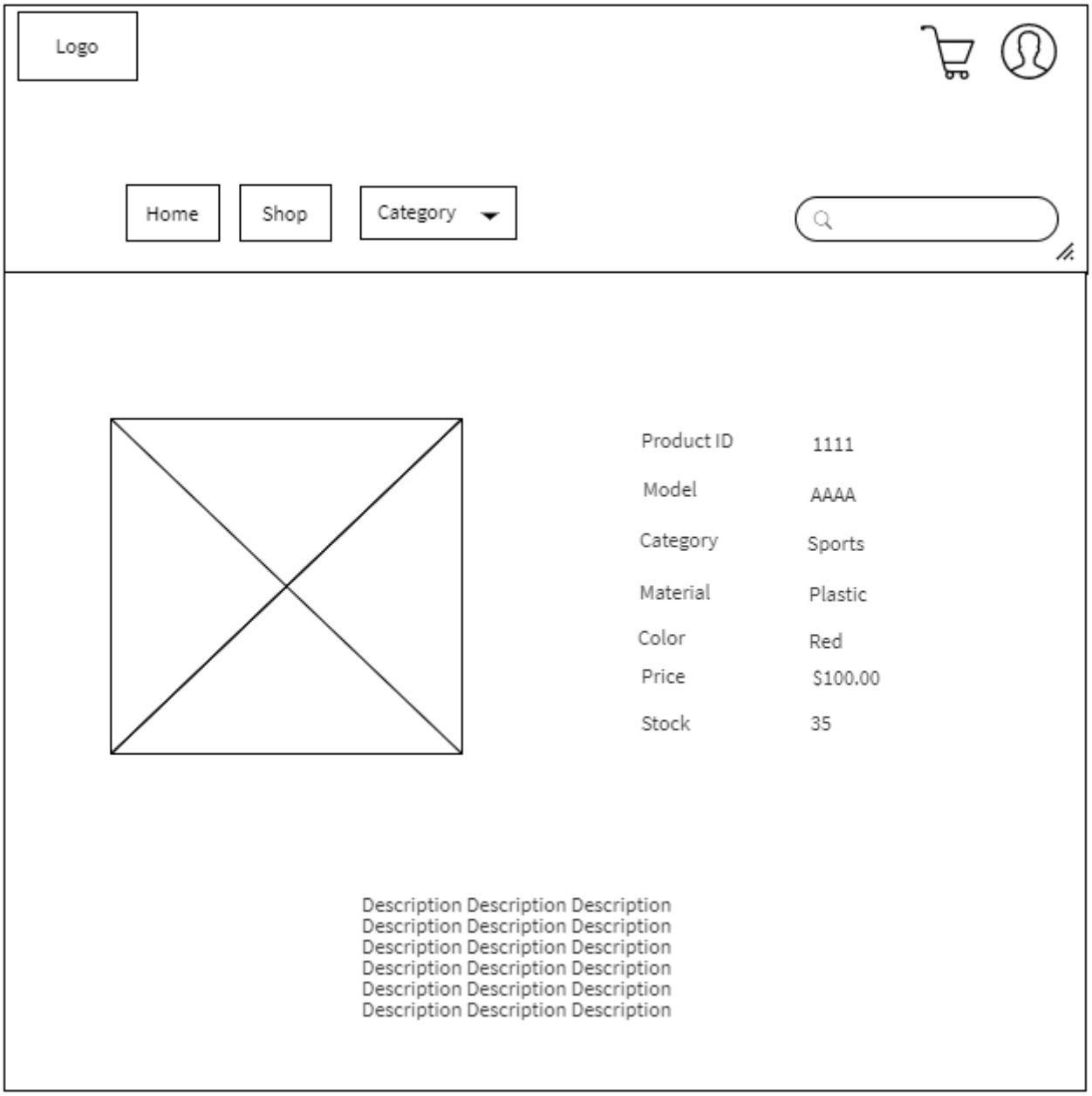
Profiling [Edit inline] [Edit] [Expl

Show all | Number of rows: 25 Filter rows: Search this table Sort

+ Options

productId	model	price
1	A1111	100.0000
2	A1111	200.0000
3	BBBB	300.0000
4	BBBB	250.0000

6.5 Product Details



Showing rows 0 - 0 (1 total, Query took 0.0005 seconds.)

```
SELECT p.productId, m.model, c.category, a.material, a.colour, p.price, p.stock, m.description, t.filepath FROM Product p INNER JOIN
Model m ON p.modelId = m.modelId INNER JOIN Category c ON p.categoryId = c.categoryId INNER JOIN Material a ON p.materialId =
a.materialId INNER JOIN productPhoto t ON p.productId = t.productId WHERE p.productId = 3
```

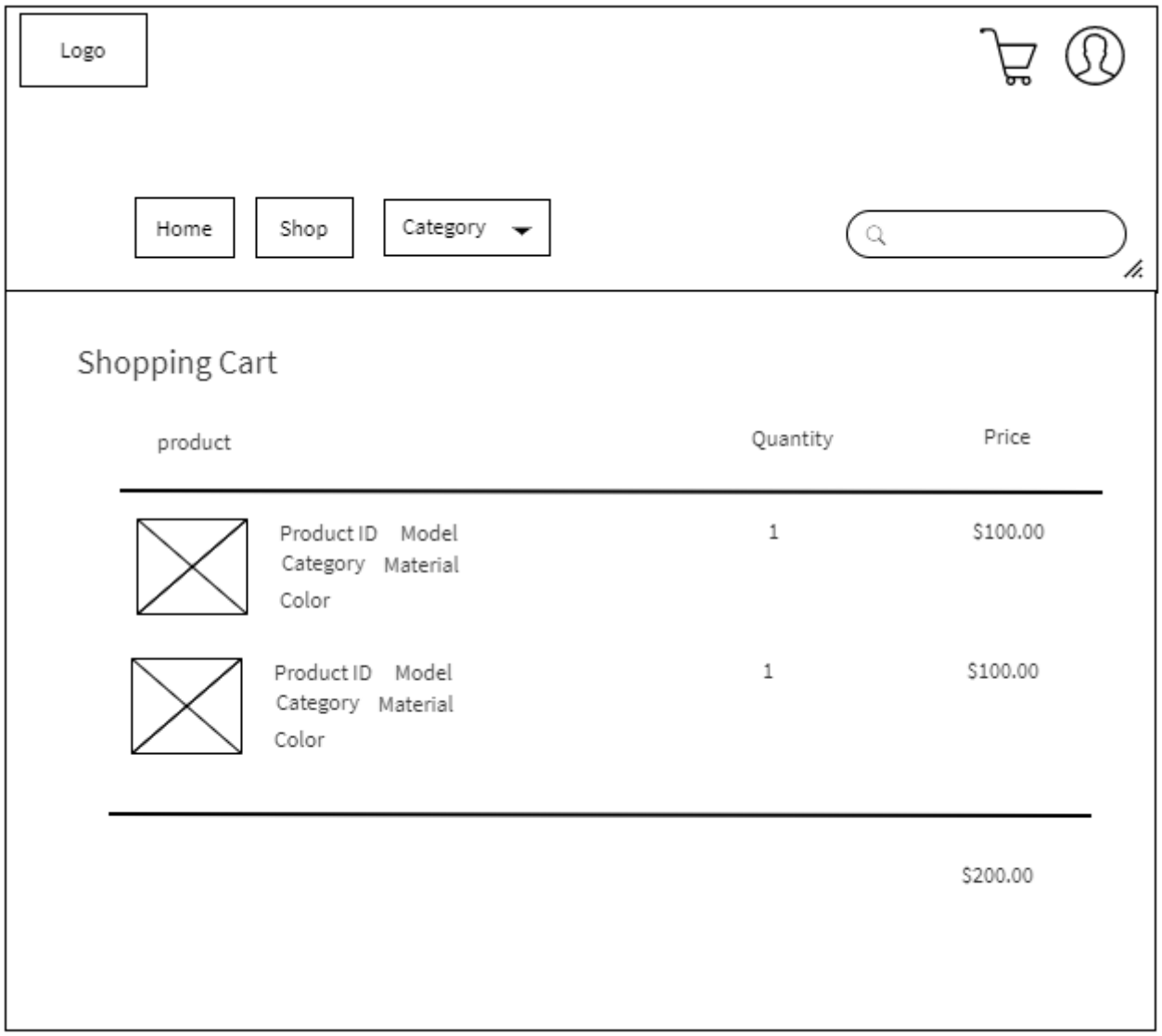
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refre

Show all | Number of rows: 25 Filter rows: Search this table

+ Options

productId	model	category	material	colour	price	stock	description	filepath
3	BBBB	sport	metal	silver	300.0000	21	sport sport	img/sunglass_model _234.jpg

6.6 Shopping Cart



Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

Showing rows 0 - 1 (2 total, Query took 0.0025 seconds.)

```
SELECT s.shopCartId, s.quantity, p.productId, m.model, c.category, a.material, a.colour, p.price, t.filepath FROM shoppingCart s INNER JOIN OrderCart oc ON
s.shopCartId = oc.shopCartId INNER JOIN Orders o ON oc.orderId = o.orderId INNER JOIN product p ON s.productID = p.productID INNER JOIN Model m ON p.modelId =
m.modelId INNER JOIN Category c ON p.categoryId = c.categoryId INNER JOIN Material a ON p.materialId = a.materialId INNER JOIN productPhoto t ON p.productId =
t.productId WHERE o.orderId = 1
```

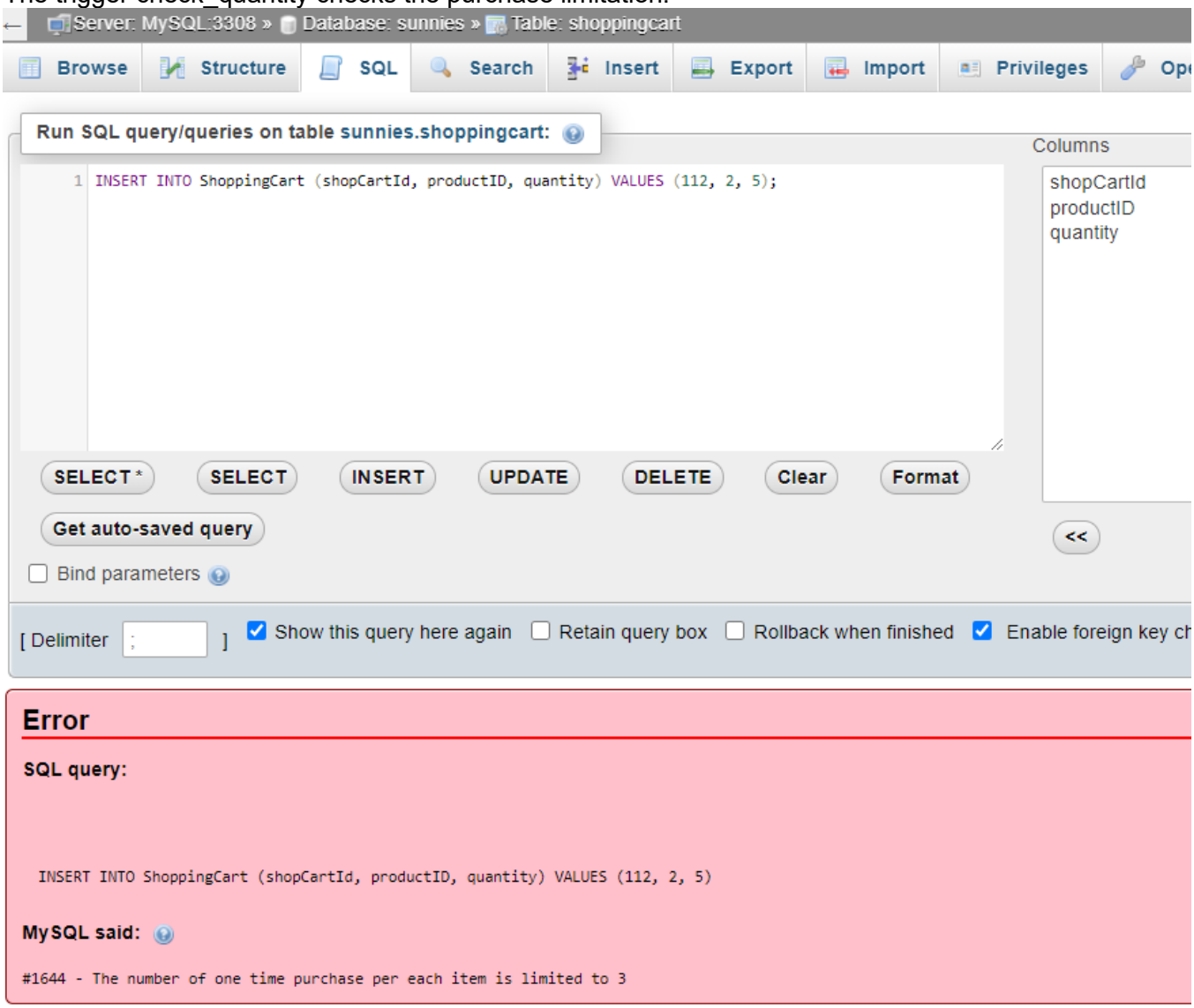
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

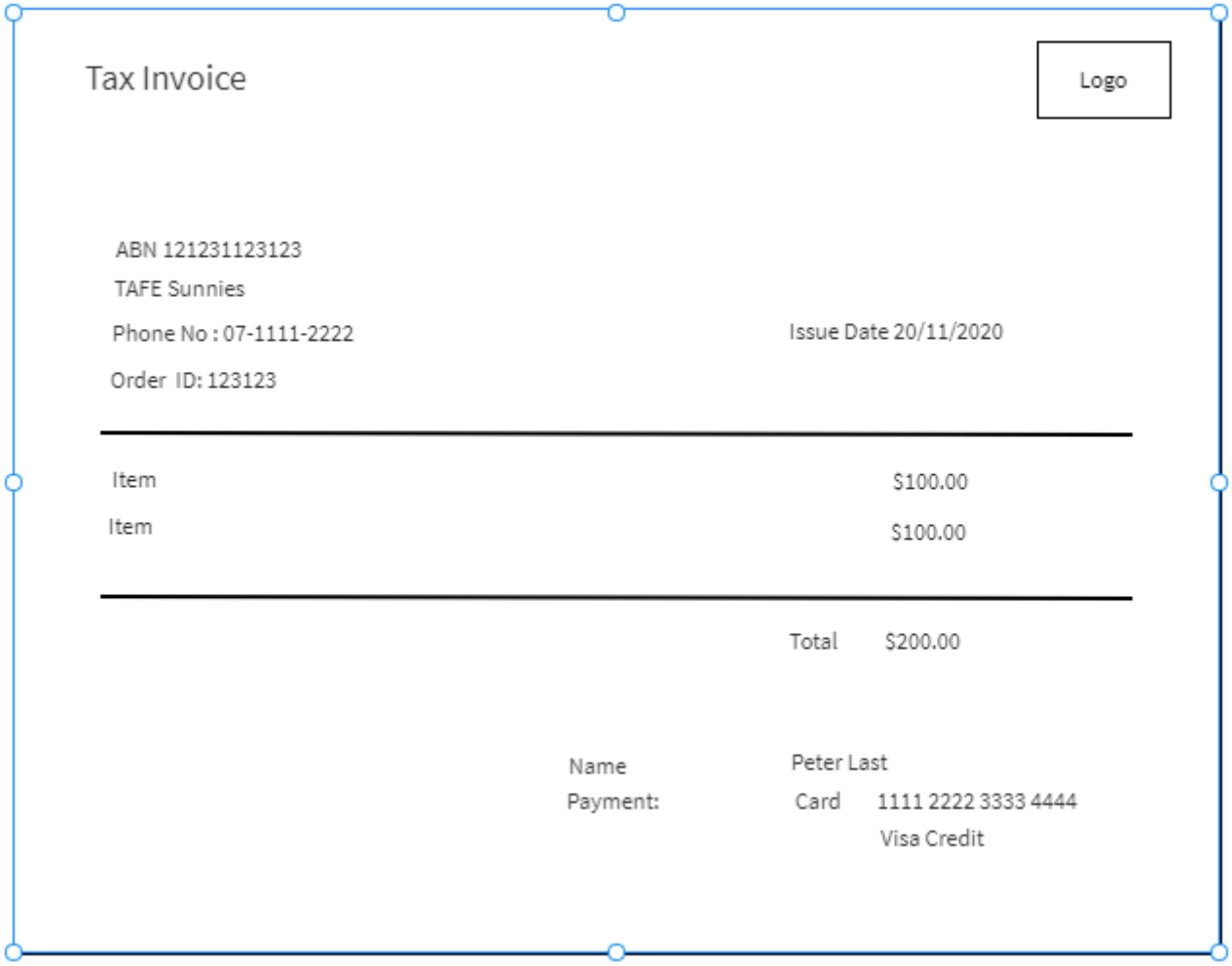
+ Options

shopCartId	quantity	productId	model	category	material	colour	price	filepath
111	1	1	A1111	sport	plastic	red	100.0000	img/sunglass_model _A1111.jpg
112	1	2	A1111	sport	metal	silver	200.0000	img/sunglass_model _234.jpg

The trigger check_quantity checks the purchase limitation.



6.7 Invoice Report



Showing rows 0 - 1 (2 total, Query took 0.0008 seconds.)

```
SELECT c.firstName, c.lastName, o.orderId, o.orderDate, o.shippedDate, d.productId, d.quantity, d.paidPrice, y.paymentType, y.paidDate, i.cardType, i.cardNo, expireDate FROM Customer c INNER JOIN Orders
o ON c.customerId = o.customerId INNER JOIN OrderDetail d ON o.orderId = d.orderId INNER JOIN product p ON d.productId = p.productId INNER JOIN Payment y ON o.paymentId = y.paymentId INNER JOIN CardInfo
i ON y.cardInfoId = i.cardInfoId WHERE o.orderId = 4
```

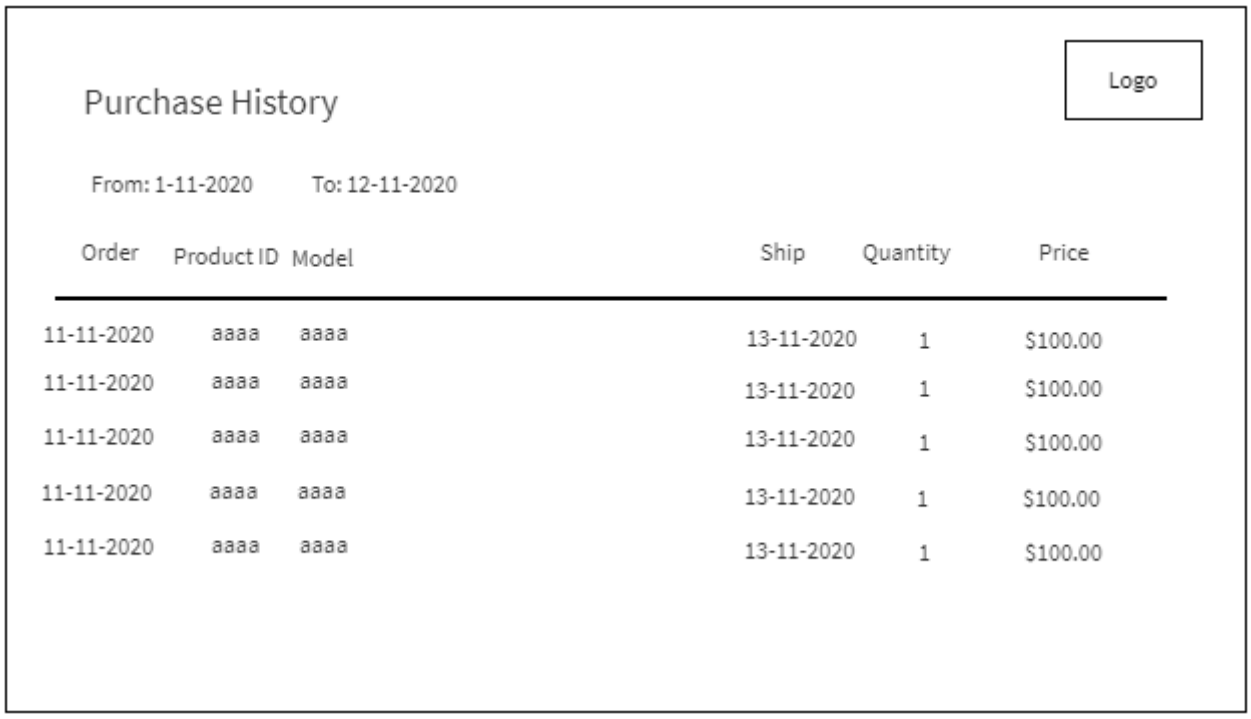
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

+ Options

firstName	lastName	orderId	orderDate	shippedDate	productId	quantity	paidPrice	paymentType	paidDate	cardType	cardNo	expireDate
Peter	Last	4	2020-11-02	2020-11-12	3	1	85.0000	card	2020-11-02	visa	1111222233334444	2023-01-01
Peter	Last	4	2020-11-02	2020-11-12	4	1	170.0000	card	2020-11-02	visa	1111222233334444	2023-01-01

6.8 Purchase History



Peter Last's purchase history between 02-11-2020 and 15-11-2020

Showing rows 0 - 3 (4 total, Query took 0.0007 seconds.)

```
SELECT c.firstName, c.lastName, o.orderId, o.orderDate, o.shippedDate, d.productId, m.model, d.quantity, d.paidPrice FROM Customer c
INNER JOIN Orders o ON c.customerId = o.customerId INNER JOIN OrderDetail d ON o.orderId = d.orderId INNER JOIN product p ON d.productId
= p.productId INNER JOIN model m ON p.modelId = m.modelId WHERE o.customerId = 1 AND o.orderDate > '2020-11-01' AND o.orderDate < '2020-
11-16'
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

+ Options

firstName	lastName	orderId	orderDate	shippedDate	productId	model	quantity	paidPrice
Peter	Last	1	2020-11-11	2020-11-14	1	A1111	1	85.0000
Peter	Last	1	2020-11-11	2020-11-14	2	A1111	1	170.0000
Peter	Last	4	2020-11-02	2020-11-12	3	BBBB	1	85.0000
Peter	Last	4	2020-11-02	2020-11-12	4	BBBB	1	170.0000

All purchase history between 02-11-2020 and 15-11-2020

Showing rows 0 - 5 (6 total, Query took 0.0008 seconds.)

```
SELECT c.firstName, c.lastName, o.orderId, o.orderDate, o.shippedDate, d.productId, m.model, d.quantity, d.paidPrice FROM Customer c
INNER JOIN Orders o ON c.customerId = o.customerId INNER JOIN OrderDetail d ON o.orderId = d.orderId INNER JOIN product p ON d.productId
= p.productId INNER JOIN model m ON p.modelId = m.modelId WHERE o.orderDate > '2020-11-01' AND o.orderDate < '2020-11-16'
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

+ Options

firstName	lastName	orderId	orderDate	shippedDate	productId	model	quantity	paidPrice
Peter	Last	1	2020-11-11	2020-11-14	1	A1111	1	85.0000
Peter	Last	1	2020-11-11	2020-11-14	2	A1111	1	170.0000
Tiny	Tony	2	2020-11-15	*NULL*	3	BBBB	1	85.0000
Tiny	Tony	2	2020-11-15	*NULL*	4	BBBB	1	170.0000
Peter	Last	4	2020-11-02	2020-11-12	3	BBBB	1	85.0000
Peter	Last	4	2020-11-02	2020-11-12	4	BBBB	1	170.0000

7. Physical design

7.1 Physical Database

The exported SQL file attached

7.2 Comparison conceptual model and technical requirement

Client Requirements	ERD Conceptual model	Data Dictionary Logical Model	Database Physical model	Comments
Member Registration	Done (covered)	Covered	Covered	
15% discount	No	No	No	Implement in programming
Product Quantity < 4	no	trigger	trigger	
Limitation of under aged purchase	no	trigger	trigger	
Member Registration	Covered	Covered	Covered	
Login in	Covered	Covered	Covered	
Product registration	Covered	Covered	Covered	
Display all product	Covered	Covered	Covered	
Display product by category	Covered	Covered	Covered	
Display Shopping cart	Covered	Covered	Covered	
Invoice report	Covered	Covered	Covered	
Retrieve Permanent sales records by date range	Covered	Covered	Covered	
Retrieve Permanent sales records by customer	Covered	Covered	Covered	

7.3 Backup and Recovery plan

To backup the database, the export from phpMyAdmin will be used, and the exported file will be saved in a separated backup server.
Restoring will be done in phpMyAdmin through the import function using the backup file.

8. Security

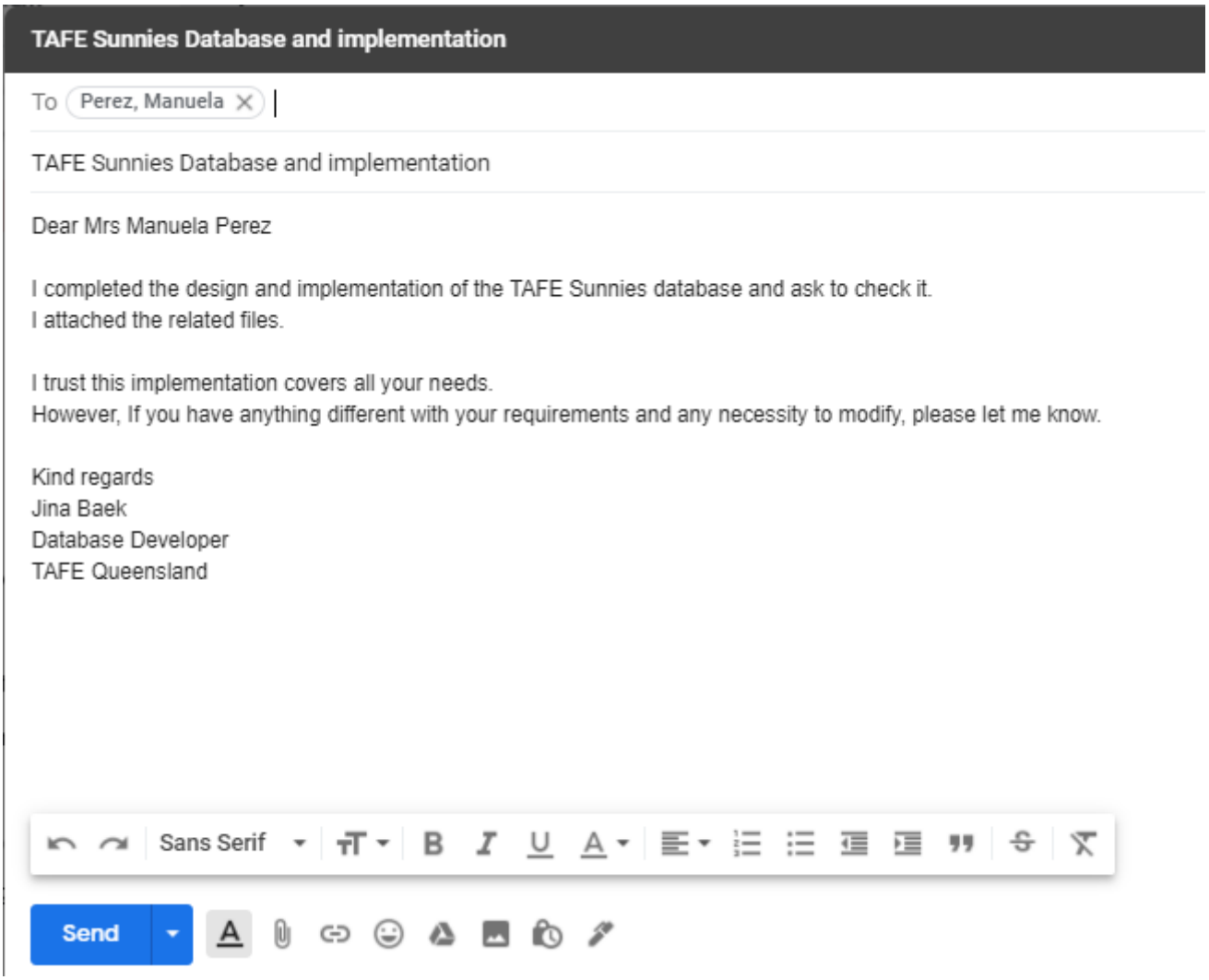
All users including admin users can access through the authentication of login ID and password. The password will be encrypted through programming in future, specifically Hashing method and session

management will be used. Therefore, the actual password data in this database implementation is not encrypted yet.

The login Id and password are saved in the Login table. When a user logs in, the application accesses the database and check the password correct.

There are three level of users, a guest, member and admin user. A guest can purchase their product without member registration. A member can retrieve their own data. All admin users can access the database through the website and manipulate the product information. The user privilege information of an admin user is saved in the Admin table. When an admin user connects the website, the application checks the Admin table and confirm the accessibility.

9. Client receipt



10. References

[1] *"Determining SQL Server Database Storage Requirements". 2020.* Searchsqlserver. *https://searchsqlserver.techtarget.com/tip/Determining-SQL-Server-database-storage-requirements#:~:text=To%20calculate%20y*

[2] *Choosing the Best Linux Distro For A Web Server. (2012). Retrieved 24 August 2020, from https://www.maketecheasier.com/choosing-the-best-linux-distro-for-a-web-server/*

[3] *MySQL :: MySQL 8.0 Reference Manual :: 6.4.1.5 PAM Pluggable Authentication. (2020). Retrieved 24 August 2020, from https://dev.mysql.com/doc/refman/8.0/en/pam-pluggable-authentication.html*

[4] website?, 1. (2008). How much disk space do you need for an ecommerce website?. Retrieved 24 August 2020, from https://ozcart.com/ideas-and-inspiration/how-much-disk-space-do-i-need-for-an-ecommerce-website#:~:text=If%20you%20have%20between%201,least%204%20GB%20of%20storage.

[5] *Hardware And Software Requirements for E-Commerce Websites - Appedology. (2020). Retrieved 24 August 2020, from https://appedology.com/hardware-and-software-requirements-for-e-commerce-websites/*