VIETNAM NATIONAL UNIVERSITY, HANOI

INTERNATIONAL SCHOOL

INS2055 – DATABASE SYSTEMS

PROJECT REPORT

Highlands Coffee's Sales Database Management

Group information

Lecturer

Trần Thị Oanh

Prepared by: Group 2

Lê Thanh Thảo

Đỗ Công Tuấn

Hanoi – June 7th, 2020

MEMBERS' CONTRIBUTION

Member name	Responsibility
Lê Thanh Thảo (100%)	Gathering data, writing business rules and relational schema, brainstorming business questions, solving SQL statement, solving the questions using My SQL, collect information, Concluding information into documents
Đỗ Công Tuấn (100%)	Designing ERD, gathering data, create database using My SQL, brainstorming business questions, solving and doing SQL statement , collect figure, interview.

Table of Notations and Abbreviations

Abbreviation	Meaning
ERD	Entity Relationship Diagram
ID	Identification

List of Tables

1. Product
2. Supplier
3. Warehouse_Entry
4. Vendor_Invoice
5. Staff
6. Order_Detail
7. Invoice
8. Table
9. Guest

List of Figures

CONTENTS

ABSTRACT/ TÓM TẮT

This report examines about the direct sales system for Highlands Coffee at the 1st floor facility, Indochina Trade Center, 241 Xuan Thuy, Cau Giay District, Hanoi. Using the direct selling process, the management system is the primary source for operating Highlands Coffee business activities. The type of products Highlands Coffee is teas, iced blended, fruit juice, cake.. Highlands Coffee need a functional database system in order to realize their aim of giving the company with the most complete system administration while also continuously going forward. Knowing that, our group has conducted an in-depth research to have a deeper understanding on their operation, as well as giving an attempt to design a Database System for this cafe based on the knowledge that we have picked up from studying this subject.

The database system will support the books as well as a few other products orders, current status of them, guest invoice, guests and supplier information. The database maintained by this system usually contains the order details, products details, supplier data, staff information, warehouse entry and guest details

We used SQL Server to connect to database and perform various operations. We assessed departmental needs, identified records, and researched database. We developed database-based application to meet current requirements and allow further expandability. Highlands Coffee would therefore benefit from this centralized database management.

Chapter 1: INTRODUCTION

1. About the Organization

Referring to the famous and long-standing coffee shops in Vietnam, it is impossible couldn't mention Highlands Coffee. Highlands Coffee has grown popular and trusted with Vietnamese people over the years for coffee connoisseurs. Highlands Coffee was founded by David Thai in 1999 with the goal of conveying a sense of pride and a harmonious relationship between tradition and modernity. In 1998, David Thai concentrated only on growing the packaged coffee business, in 2002, the first Highlands Coffee was officially launched with the two main colors of red and black of the cafe. Nowadays, Highlands Coffee has more than 200 establishments throughout Vietnam.

Staff here are trained to bring exquisite and attentive services to guests. In addition, the cafe also has many attractive deals on special occasions. For example: Exchange 1 VinID point for a buy 1 get 1 voucher , discounts, vouchers.. Besides traditional coffee such as Vietnamese PHIN coffee, iced black coffee, highlands Coffee also creates many new products as Freeze or PhinDi. Going to the café and yet being able to taste a selection of teas is a unique experience. Green tea, peach jelly tea, and tea with lotus seeds are all appealing. The price range in the Highlands is extremely comfortable, ranging from 30,000 to 70,000 VND. Suitable for a variety of audiences

For Highlands Coffee on the first floor, Indochina Trade Center, 241 Xuan Thuy, the space is quite comfortable. A special feature of here is that it is located opposite Hanoi National University, so there are quite a few students coming to study. In addition, Cau Giay district is also the location of large companies, so commuters often come here to discuss business.

2. Requirement Specification

2.1. Methods for investigating business rules:

i. Interview

Our crew has gained insight into Highlands Coffee's sales operation thanks to a acquaintance who works in this organization at the site where we are working on the project. So, while I was in quarantine due to Covid-19, I conducted a Facebook Messenger interview with a friend who is the cafe's manager. Through the interview, I understood the work flow they use to manage the incoming and outgoing orders as well as the goods in their store. The following is a narration of an interview recorded by our team:
https://drive.google.com/drive/folders/1oKqi-0OYX0NptCqFcAepaLqAEQbagL0f?usp=sharing

Reading news/websites

Highlands Coffee also run an online website that allows guests to view locations across cities, product information on their menu, etc. without having to visit their store directly. Hence, having a glance through this website as a reference has helped us to sketch in details how they categorized and group the products on their system: https://www.highlandscoffee.com.vn/

In addition, my team also found information about Highlands Coffee through the following websites: https://vi.wikipedia.org/wiki/Highlands_Coffee,
https://www.facebook.com/highlandscoffeevietnam

https://vinid.net/blog/review-highlands-coffee-menu-bang-gia-san-pham-uu-dai/

ii. Observation

Our group's members are all guests of Highlands Coffee so it is easy for us to invest the information and also after the interview with Highland manager, we could following information can be inferred:

Should reduce the volume of notebooks to store information in computer system

Capable of storing long-term information, ensuring quick query

3. Business Narrative

Every product from Highlands Coffee has a ProductID and other information is stored

This is the process of direct sales system at Highlands Coffee

Process 1: Warehouse

- Product Claim: When the products run out, staff will notify the manager to import. The manager will contact the suppliers to request the quantity of products. A supplier information is kept includes: Supplier ID, Company name, Supplier address, Supplier phone.

- Examine products: When new product are imported to the store, the bartender is in charge of inspecting the quantity of products and informing the cashier to recorded in the system.

- Warehouse entry: Imported products will be store in a Warehouse_Entry on the system for convenient tracking of products and drafting vendor invoices

- Create vendor invoice: When the product is entered on the Warehouse_Entry, the cashier will create a Vendor_Invoice to transmit to the manager, who then emails it to accounting department

Process 2: Direct sales

- Products order: Based on Highlands Coffee's menu, guests will order products according to their needs and recorded by the cashier on the Order_Detail. If guests has promotion code, the cashier will fill out the promotion on the Order_Detail.

Process 3: Payment

- Create invoices: The cashier will create an Invoice based on the actual quantity that the guests has ordered in the Order_Detail and collect the exact amount that they has to pay.

There are two options for payment: Cash / Credit Card

Guest can choose between two options for ways for cashier to draft the Invoice.

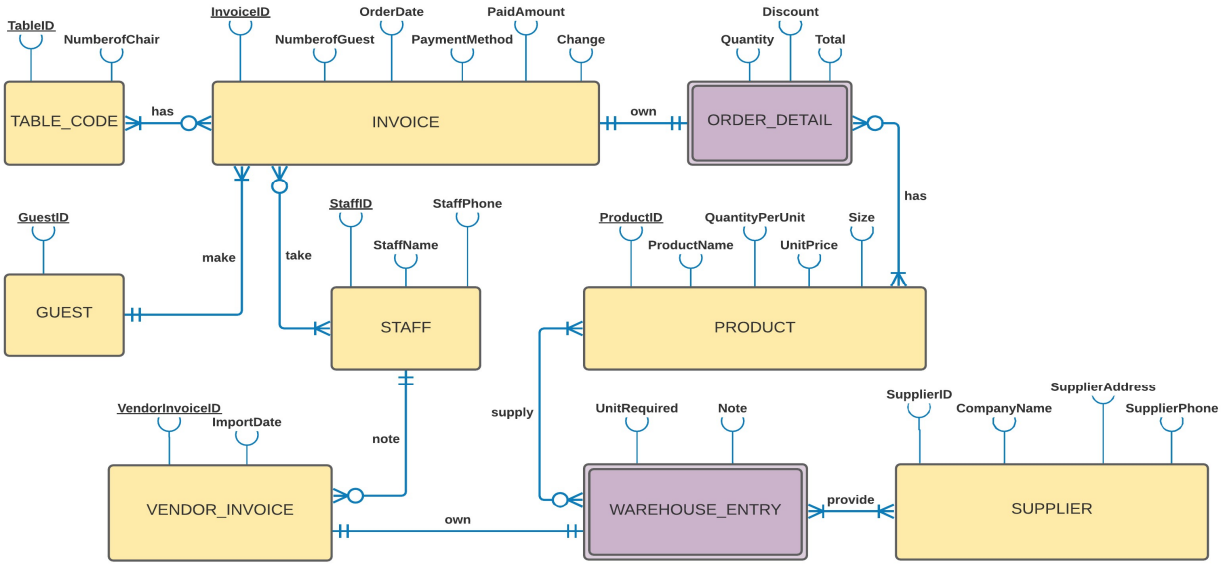
- Table code: Each Invoice will be associated with a Table_Code on the system. After payment, guests will receive a table code in the form of a vibrator and an invoice to receive the products. When products is done, the table code will ring and the guest will pick up at the order counter.

→ Finish payment and receive products

In order to maintain track of staff working with invoices, each person's information will be kept on the system is Staff: which include Staff ID, Staff name, Staff phone

Chapter 2: ANALYZING AND DESIGNING THE DATABASE

1. ERD



2. Relational Schema

GUEST(GuestID, InvoiceID)

PRIMARY KEY(GuestID)

FOREIGN KEY (InvoiceID) REFERENCES INVOICE(InvoiceID)

TABLE_CODE(TableID, NumberofChair, InvoiceID)

PRIMARY KEY(TableID)

FOREIGN KEY (InvoiceID) REFERENCES INVOICE(InvoiceID)

INVOICE(InvoiceID, NumberofGuest, OrderDate, PaymentMethod, PaidAmount, Change, GuestID, TableID, StaffID)

PRIMARY KEY(InvoiceID)

FOREIGN KEY (GuestID) REFERENCES GUEST(GuestID)

FOREIGN KEY (TableID) REFERENCES TABLE(TableID)

FOREIGN KEY (StaffID) REFERENCES STAFF(StaffID)

ORDER_DETAIL(InvoiceID, ProductID, Quantity, Discount, Total)

FOREIGN KEY (InvoiceID) REFERENCES INVOICE(InvoiceID)

FOREIGN KEY (ProductID) REFERENCES PRODUCT(ProductID)

PRODUCT(ProductID, ProductName, QuantityPerUnit, UnitPrice, Size)

PRIMARY KEY(ProductID)

WAREHOUSE_ENTRY(ProductID, SupplierID, VendorInvoiceID, UnitRequired, Note)

FOREIGN KEY (ProductID) REFERENCES PRODUCT(ProductID)

FOREIGN KEY (SupplierID) REFERENCES SUPPLIER(SupplierID)

FOREIGN KEY (VendorInvoiceID) REFERENCES VENDOR_INVOICE(VendorInvoiceID)

SUPPLIER(SupplierID, CompanyName, SupplierAddress, SupplierPhone)

PRIMARY KEY(SupplierID)

VENDOR_INVOICE(VendorInvoiceID, ImportDate, StaffID)

PRIMARY KEY(VendorInvocieID)

FOREIGN KEY (StaffID) REFERENCES STAFF(StaffID)

STAFF(StaffID, StaffName, VendorInvoiceID, InvoiceID)

PRIMARY KEY(StaffID)

FOREIGN KEY (VendorInvoiceID) REFERENCES VENDOR_INVOICE(VendorInvoiceID)

FOREIGN KEY (InvoiceID) REFERENCES INVOICE(InvoiceID)

3. Some real data and SQL scripts

Here is some sql script data of PRODUCT table

```
CREATE TABLE Product (
ProductID INT(10) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
ProductName VARCHAR(30) NOT NULL,
Size VARCHAR(5),
QuantityPerUnit INT NOT NULL,
UnitPrice INT NOT NULL
);
INSERT INTO Product(ProductName, Size, QuantityPerUnit, UnitPrice)
    VALUE   ('Phin sữa đá', '12oz', 243, 35000),        ('Phin sữa đá', '16oz', 331, 39000),
            ('Bạc xỉu đá', '10oz', 12, 29000),          ('Phindi kem sữa', '12oz', 76, 45000),
            ('Phindi hạnh nhân', '12oz', 28, 45000),    ('Phindi choco', '16oz', 143, 49000),
            ('Bánh Pasty', '', 32, 29000),              ('Bánh chuối', '', 11, 29000),
            ('Mousse cacao', '', 48, 29000),            ('Trà sen vàng', '12oz', 196, 39000),
            ('Trà xanh đậu đỏ', '16oz', 148, 49000),    ('Trà sen vàng', '16oz', 175, 49000),
            ('Trà Thanh Đào', '12oz', 29, 39000),       ('Phô mai trà xanh', '', 37, 29000),
            ('Phô mai caramel', '', 44, 29000),         ('Freeze chocolate', '12oz', 199, 49000),
```

Here is some sql script data of SUPPLIER table

```
CREATE TABLE Supplier (
SupplierID INT(10) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
CompanyName VARCHAR(30) NOT NULL,
SupplierPhone VARCHAR(13) NOT NULL,
SupplierAddress VARCHAR(30)
);
INSERT INTO Supplier(CompanyName, SupplierPhone, SupplierAddress)
    VALUE   ('Nguyễn Sơn', '0969454545', 'Bắc Từ Liêm - Hà Nội'),
            ('Mega Market', '02253528128', 'Bắc Từ Liêm - Hà Nội'),
            ('Kim Cương', '0963131419', 'Hai Bà Trưng - Hà Nội'),
            ('DalatMilk', '0901778775', 'Cầu Giấy - Hà Nội'),
            ('Thành Đạt', '0904192388', 'Hoàng Mai - Hà Nội'),
            ('An Phú', '0982636618', 'Ba Vì - Hà Nội'),
            ('Abby', '1900779907', 'Hà Đông - Hà Nội'),
            ('ĐVP Market', '0907309988', 'Bắc Từ Liêm - Hà Nội'),
            ('Nhất An', '0931847626', 'Ba Vì - Hà Nôi'),
```

Here is some sql script data of WAREHOUSE_ENTRY table

```
CREATE TABLE WarehouseEntry (
VendorInvoiceID INT(10) UNSIGNED, FOREIGN KEY (VendorInvoiceID) REFERENCES VendorInvoice(VendorInvoiceID),
ProductID INT(10) UNSIGNED, FOREIGN KEY (ProductID) REFERENCES Product(ProductID),
UnitRequired INT NOT NULL,
SupplierID INT(10) UNSIGNED, FOREIGN KEY (SupplierID) REFERENCES Supplier(SupplierID),
Note VARCHAR(100)
);
INSERT INTO WarehouseEntry(VendorInvoiceID, ProductID, UnitRequired, SupplierID, Note)
    VALUE   (1, 5, 200, 6, ''), (1, 2, 200, 7, ''), (1, 4, 200, 2, ''),
            (2, 1, 200, 1, ''), (2, 6, 200, 4, ''), (2, 9, 200, 10, ''),
            (3, 14, 200, 2, ''),(3, 11, 200, 3, ''),(3, 20, 500, 6, ''),
            (4, 17, 150, 9, ''),(4, 15, 350, 10, ''),
            (5, 13, 200, 4, ''),(5, 15, 10, 4, 'Change for new ones'),
            (6, 7, 150, 3, ''), (6, 18, 150, 7, ''),
            (7, 14, 300, 2, ''),(7, 11, 350, 2, ''),
            (8, 6, 300, 8, 'Prepare for Liberation Day'),   (8, 4, 500, 9, 'Prepare for Liberation Day'),
```

Here is some sql script data of VENDOR_INVOICE table

```
CREATE TABLE VendorInvoice (
VendorInvoiceID INT(10) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
StaffID INT(10) UNSIGNED, FOREIGN KEY (StaffID) REFERENCES Staff(StaffID),
ImportDate DATE NOT NULL
);
INSERT INTO VendorInvoice(StaffID, ImportDate) VALUE    (2, '2021-01-19'),
                                                        (1, '2021-01-31'),
                                                        (3, '2021-02-16'),
                                                        (6, '2021-02-28'),
                                                        (7, '2021-03-01'),
                                                        (8, '2021-03-15'),
                                                        (10, '2021-04-01'),
                                                        (2, '2021-04-19'),
                                                        (6, '2021-05-02'),
                                                        (4, '2021-05-24');
```

Here is some sql script data of STAFF table

```sql
CREATE TABLE Staff (
StaffID INT(10) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
StaffName VARCHAR(50) NOT NULL,
StaffPhone VARCHAR(13)
);
INSERT INTO Staff(StaffName, StaffPhone) VALUE ('Trịnh Tường Vi', '0869289098'),
                                                ('Hoàng Trung Kiên', '0372115628'),
                                                ('Cao Đức Đạt', '0963425789'),
                                                ('Tống Thu Hà', '0320672178'),
                                                ('Nguyễn Trường Giang', '0392774891'),
                                                ('Đỗ Xuân Quyền', '0912325841'),
                                                ('Phạm Nhật Kim Anh', '0331547347'),
                                                ('Vũ Thùy Linh', '0398298345'),
                                                ('Vũ Quỳnh Trang', '0979809244'),
                                                ('Trần Thị Kiều Linh', '0357213743');
```

Here is some sql script data of INVOICE table

```sql
CREATE TABLE Invoice (
InvoiceID INT(10) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
GuestID INT(10) UNSIGNED, FOREIGN KEY (GuestID) REFERENCES Guest(GuestID),
StaffID INT(10) UNSIGNED, FOREIGN KEY (StaffID) REFERENCES Staff(StaffID),
TableID INT(2) UNSIGNED, FOREIGN KEY (TableID) REFERENCES TableCode(TableID),
NumberOfGuest INT(2) UNSIGNED NOT NULL,
OrderDate DATE NOT NULL,
PaymentMethod VARCHAR(10) NOT NULL,
PaidAmount INT NOT NULL,
TheChange INT
);
INSERT INTO Invoice(GuestID, StaffID, TableID, NumberOfGuest, OrderDate, PaymentMethod, PaidAmount, TheChange)
    VALUE   (1, 3, 2, 1, '2021-04-27', 'cash', 30000, 1000),
            (2, 2, 2, 1, '2021-04-30', 'cash', 50000, 1000),
            (3, 6, 4, 1, '2021-04-30', 'cash', 100000, 25000),
            (4, 9, 5, 2, '2021-04-30', 'cash', 120000, 8000),
```

Here is some sql script data of GUEST table

```sql
CREATE TABLE Guest (
GuestID INT(10) UNSIGNED PRIMARY KEY
);
INSERT INTO Guest(GuestID) Value(1),(2),(3),(4),(5),(6),(7),(8),(9),(10),(11),(12),
                                (13),(14),(15),(16),(17),(18),(19),(20),(21),(22),(23),(24);
```

Here is some sql script data of ORDER_DETAIL table

```
CREATE TABLE OrderDetail (
InvoiceID INT(10) UNSIGNED, FOREIGN KEY (InvoiceID) REFERENCES Invoice(InvoiceID),
ProductID INT(10) UNSIGNED, FOREIGN KEY (ProductID) REFERENCES Product(ProductID),
UnitPrice INT NOT NULL,
Quantity INT NOT NULL,
Discount INT NOT NULL,
Total INT AS ((UnitPrice*Quantity) - Discount)
);
INSERT INTO OrderDetail(InvoiceID, ProductID, UnitPrice, Quantity, Discount)
    VALUE   (1, 3, 29000, 1, 0 ),
            (2, 11, 49000, 1, 0),
            (3, 13, 39000, 2, 3000),
            (4, 13, 29000, 2, 1000),    (4, 17, 55000, 1, 0),
            (5, 1, 35000, 1, 0),        (5, 19, 55000, 1, 0),
            (6, 8, 29000, 2, 3000),     (6, 20, 9000, 20, 8000),
            (7, 10, 39000, 1, 0),       (7, 12, 49000, 1, 1000),
            (8, 18, 39000, 1, 0),       (8, 4, 45000, 1, 0),
            (9, 16, 49000, 1, 0),       (9, 6, 49000, 2, 3000),
            (9, 5, 45000, 1, 0),
```

Here is some sql script data of TABLE_CODE table

```
/*---TABLE_CODE TABLE---*/
CREATE TABLE TableCode (
TableID INT(2) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
NumberOfChair INT(1) NOT NULL
);
INSERT INTO TableCode(NumberOfChair) VALUE (1), (1), (1), (1), (1),
                                           (2), (2), (2), (2), (2),
                                           (4), (4), (4), (4), (4),
                                           (2), (2), (2), (2), (2);
```

Chapter 3: SOME USEFUL REPORTS AND INFORMATION

1. Design some useful reports and information for real usage

2. Based on the designed database for Highlands Coffee's direct sales management, we assign some questions to test the effective of the system as well as to manage and set promotion for improvement.

List of 10 questions:

1. Manager the history of guests payment
2. At a specific date, what is the amount of each product was sold?
3. What is the top 5 Best-Seller products?
4. The difference of days between today and the latest supply day
5. Total number of products and their revenue on Reunification Day and International Workers' Day
6. Manage total revenue with payment method
7. Supply of product 'Chanh dây đá viên" has arrived. And now should the database with this new one. Show the new-added Product on the screen (Transaction in used)
8. The data of a product has changed and the manager want to manage the history of the change (Trigger in used)
9. The manager wants to print out all the vendor invoice of 20/04 to 05/05 (Procedure in used)
10. Today 05/05/2021. We arrive at work and discover that our new data entry clerk in training has entered all new Invoice incorrectly on the last. We want to teach our trainee to find and correct all erroneous records. What's the easiest way to get all the records from the Invoice table entered on the last?

3. Solutions for the above reports/information

Question 1: Manager the history of guests payment

```
CREATE VIEW PaidHistory AS
SELECT Guest.GuestID, InvoiceID, OrderDate, PaymentMethod, PaidAmount, TheChange
FROM Invoice
INNER JOIN Guest ON Guest.GuestID = Invoice.GuestID
ORDER BY OrderDate ASC;
                        SELECT * FROM PaidHistory;
```

GuestID	InvoiceID	OrderDate	PaymentMethod	PaidAmount	TheChange
1	1	2021-04-27	cash	30000	1000
2	2	2021-04-30	cash	50000	1000
3	3	2021-04-30	cash	100000	25000
4	4	2021-04-30	cash	120000	8000
5	5	2021-04-30	cash	100000	10000
6	6	2021-04-30	cash	150000	13000

Question 2: At a specific date, what is the amount of each product was sold?

```
CREATE VIEW SaleAtDate AS
SELECT Pro.ProductID, Pro.ProductName, Pro.Size,
SUM(Ordd.Quantity) as Quantity, Invoice.OrderDate AS SOLD_DATE
FROM ((OrderDetail Ordd
INNER JOIN Product Pro ON Pro.ProductID = Ordd.ProductID)
INNER JOIN Invoice ON Ordd.InvoiceID = Invoice.InvoiceID)
GROUP BY SOLD_DATE, ProductID
ORDER BY SOLD_DATE ASC;

            SELECT * FROM SaleAtDate;
```

ProductID	ProductName	Size	Quantity	SOLD_DATE
3	Bạc xỉu đá	10oz	1	2021-04-27
1	Phin sữa đá	12oz	1	2021-04-30
8	Bánh chuối		2	2021-04-30
11	Trà xanh đậu đỏ	16oz	1	2021-04-30
13	Trà Thanh Đào	12oz	4	2021-04-30
17	Chanh dây đá viên	16oz	1	2021-04-30
19	Chocolate		1	2021-04-30
20	Thạch vải		20	2021-04-30

Question 3: What is the top 5 Best-Seller products?

```
CREATE VIEW BestSeller AS
SELECT Pro.ProductID, Pro.ProductName, Pro.Size, Pro.UnitPrice,
       Ordd.Discount, SUM(Ordd.Quantity) AS Sold_out, Ordd.Total AS Profit
FROM Product Pro
INNER JOIN OrderDetail Ordd ON Pro.ProductID = Ordd.ProductID
GROUP BY ProductID
ORDER BY Total DESC
LIMIT 5;
```

```
SELECT * FROM BestSeller;
```

ProductID	ProductName	Size	UnitPrice	Discount	Sold_out	Profit
15	Phô mai caramel		29000	11000	9	250000
20	Thạch vải		9000	8000	31	172000
9	Mousse cacao		29000	6000	6	168000
6	Phindi choco	16oz	49000	3000	9	95000
14	Phô mai trà xanh		29000	2000	3	85000

Question 4: The difference of days between today and the latest supply day

```
CREATE VIEW DayDiff AS
SELECT GR.VendorInvoiceID, EI.StaffID, Supplier.SupplierID, Supplier.CompanyName, GR.Note,
       EI.ImportDate AS Lastest_Supply_Day, CURDATE() AS Today,
       DATEDIFF(CURDATE(), EI.ImportDate) AS Day_Interval
FROM ((WarehouseEntry GR
INNER JOIN VendorInvoice EI ON EI.VendorInvoiceID = GR.VendorInvoiceID)
INNER JOIN Supplier ON Supplier.SupplierID = GR.SupplierID)
ORDER BY VendorInvoiceID DESC
LIMIT 1;
```

```
SELECT * FROM DayDiff;
```

VendorInvoiceID	StaffID	SupplierID	CompanyName	Note	Lastest_Supply_Day	Today	Day_Interval
10	4	6	An Phú		2021-05-24	2021-06-04	11

Question 5: Total number of products and their revenue on Reunification Day and International Workers' Day

```
CREATE VIEW SaleOn5D1 AS
SELECT Pro.ProductID, Pro.ProductName, Invoice.OrderDate AS Sold_date,
       SUM(Ordd.Quantity) AS Sold_out, SUM(Ordd.Discount) AS Total_discount, Ordd.Total AS Revenue
FROM ((OrderDetail Ordd
INNER JOIN Product Pro ON Pro.ProductID = Ordd.ProductID)
INNER JOIN Invoice ON Invoice.InvoiceID = Ordd.InvoiceID)
WHERE OrderDate = '2021-05-01' OR OrderDate = '2021-04-30'
GROUP BY ProductID
ORDER BY Sold_date ASC;
                              SELECT * FROM SaleOn5D1;
```

ProductID	ProductName	Sold_date	Sold_out	Total_discount	Revenue
11	Trà xanh đậu đỏ	2021-04-30	1	0	49000
13	Trà Thanh Đào	2021-04-30	6	7000	75000
17	Chanh dây đá viên	2021-04-30	1	0	55000
1	Phin sữa đá	2021-04-30	1	0	35000

Question 6: Manage total revenue with payment method

```
CREATE VIEW RevPayMethod AS
SELECT Invoice.PaymentMethod, COUNT(DISTINCT(Invoice.InvoiceID)) AS Number_Of_Invoice,
       SUM(Ordd.Quantity) AS Sold_out, SUM(Ordd.Total) AS Revenue
FROM OrderDetail Ordd
INNER JOIN Invoice ON Invoice.InvoiceID = Ordd.InvoiceID
GROUP BY PaymentMethod
ORDER BY Revenue DESC;
                    SELECT * FROM RevPayMethod;
```

PaymentMethod	Number_Of_Invoice	Sold_out	Revenue
cash	18	74	2018000
creditcard	6	42	1443000

Question 7: Supply of product ‘Chanh dây đá viên” has arrived. And now should update the database with this new one. Show the new-added Product on the screen (Transaction in used)

```
ALTER TABLE Product ADD CHECK (QuantityPerUnit <= 1500);
/*Transaction will rollback when QuantityPerUnit > 1500 because of the maximum storage's capaity*/

START TRANSACTION;
SET AUTOCOMMIT = ON;
UPDATE Product
SET QuantityPerUnit = QuantityPerUnit + 300
WHERE ProductID = 17;
INSERT INTO VendorInvoice(StaffID, ImportDate) VALUE (1, CURDATE() );
                          /*Only Date is inserted, time is not included*/
INSERT INTO WarehouseEntry(VendorInvoiceID, ProductID, UnitRequired, SupplierID, Note)
        SELECT MAX(VendorInvoiceID) + 1, 17, 300, 10, 'New Supply' FROM WarehouseEntry;
            /* Insert new InvoiceID = Max(InvoiceID) + 1 */
COMMIT;

/*Display on the screen*/
SELECT GR.VendorInvoiceID, GR.ProductID, Pro.ProductName, Pro.QuantityPerUnit, Supplier.CompanyName, GR.Note
FROM ((WarehouseEntry GR
INNER JOIN Product Pro ON Pro.ProductID = GR.ProductID)
INNER JOIN Supplier ON Supplier.SupplierID = GR.SupplierID)
ORDER BY VendorInvoiceID DESC
LIMIT 1;
```

VendorInvoiceID	ProductID	ProductName	QuantityPerUnit	CompanyName	Note
12	17	Chanh dây đá viên	600	United Vision	New Supply

VendorInvoiceID	ProductID	ProductName	QuantityPerUnit	CompanyName	Note
13	17	Chanh dây đá viên	900	United Vision	New Supply

Question 8: The data of a product has changed and the manager want to manage the history of the change (Trigger in used)

```
CREATE TABLE HistoryProduct (
ProductHistoryID INT(10) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
ProductID INT(10) UNSIGNED NOT NULL,
ProductName VARCHAR(30),
Size VARCHAR(5),
QuantityPerUnit INT,
UnitPrice INT,
ChangeDate DATE NOT NULL
);

/*USE TRIGGER*/
CREATE TRIGGER Before_change_product
BEFORE UPDATE ON Product
FOR EACH ROW
    INSERT INTO HistoryProduct
        SET     ProductID = OLD.ProductID,
                ProductName = OLD.ProductName,
                Size = OLD.Size,
                QuantityPerUnit = OLD.QuantityPerUnit,
                UnitPrice = OLD.UnitPrice,
                ChangeDate = CURDATE();
/*DROP TRIGGER Before_change_product;*/

/*Demo with Thạch vải*/
UPDATE Product
SET ProductName = 'Thạch vải nhập khẩu',
   QuantityPerUnit = 500,
   UnitPrice = 10000
WHERE ProductName = 'Thạch vải';
```

ProductHistoryID	ProductID	ProductName	Size	QuantityPerUnit	UnitPrice	ChangeDate
1	20	Thạch vải		368	9000	2021-06-04
NULL	NULL	NULL	NULL	NULL	NULL	NULL

ProductID	ProductName	Size	QuantityPerUnit	UnitPrice
17	Chanh dây đá viên	16oz	900	55000
18	Tắc quất đá viên	12oz	11	39000
19	Chocolate		136	55000
20	Thạch vải nhập khẩu		500	10000

Question 9: The manager wants to print out all the vendor invoice of 20/04 to 05/05 (Procedure in used)

```
USE `highland`;
DROP procedure IF EXISTS `ManageDiscount`;
DELIMITER $$
USE `highland`$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `PrintVendorInvoice`()
BEGIN
    SELECT GR.VendorInvoiceID, GR.ProductID, Pro.ProductName, Pro.Size, Pro.QuantityPerUnit, Pro.UnitPrice,
    (Pro.QuantityPerUnit * UnitPrice) AS TotalVendor, VI.ImportDate,
    Supplier.CompanyName, GR.Note
    FROM (((WarehouseEntry GR
    INNER JOIN Product Pro ON Pro.ProductID = GR.ProductID)
    INNER JOIN Supplier ON Supplier.SupplierID = GR.SupplierID)
    INNER JOIN VendorInvoice VI ON VI.VendorInvoiceID = GR.VendorInvoiceID)
    WHERE VI.ImportDate BETWEEN '2021-04-20' AND '2021-05-25'
    ORDER BY VendorInvoiceID ASC;

END$$
DELIMITER ;

CALL PrintVendorInvoice();
```

VendorInvoiceID	ProductID	ProductName	Size	QuantityPerUnit	UnitPrice	TotalVendor	ImportDate	CompanyName	Note
9	1	Phin sữa đá	12oz	243	35000	8505000	2021-05-02	United Vision	
10	2	Phin sữa đá	16oz	331	39000	12909000	2021-05-24	An Phú	

Question 10: Today 05/05/2021. We arrive at work and discover that our new data entry clerk in training has entered all new Invoice incorrectly on the last. We want to teach our trainee to find and correct all erroneous records. What's the easiest way to get all the records from the Invoice table entered on the last?

```
/*See erroneous records*/
SELECT * FROM Invoice
WHERE OrderDate BETWEEN '2021-05-03' AND '2021-05-05';
/*Update*/
UPDATE Invoice
SET     TableID = TableID - 1
WHERE TableID > 1 AND OrderDate BETWEEN '2021-05-03' AND '2021-05-05';
```

InvoiceID	GuestID	StaffID	TableID	NumberOfGuest	OrderDate	PaymentMethod	PaidAmount	TheChange
16	16	9	16	2	2021-05-03	cash	110000	9000
17	17	5	13	3	2021-05-03	creditcard	290000	0
18	18	3	19	1	2021-05-03	cash	55000	0
19	19	2	2	1	2021-05-04	cash	155000	2000
20	20	10	15	3	2021-05-04	creditcard	309000	0
21	21	5	15	1	2021-05-05	creditcard	170000	0
22	22	8	20	2	2021-05-05	creditcard	250000	0

Chapter 4: CONCLUSION

This database system's role is to assist the store in saving data and information consistently, minimizing redundancy by specified categories. By creating a database system, the personnel will be able to quickly seek up information about the objects (quantity, price, supplier..). It also helps to research and define customer preferences for products in the store (coffee, tea, iced coffee, etc.) as well as to have appropriate and timely promotions to increase sales.

However, this approach is not without flaws, and when dealing with large amounts of data, it can produce mistakes.