# Chapter – 2

## Database Management System

This chapter unfolds the detailed explanation of the database system designed for shopping malls. This covers components such as ER diagram, Schema diagram, and data dictionary, along with schema creation and data insertion leading to the development of a complete database, ready to enhance retail efficiency.

### 2.1: Project Diagrams

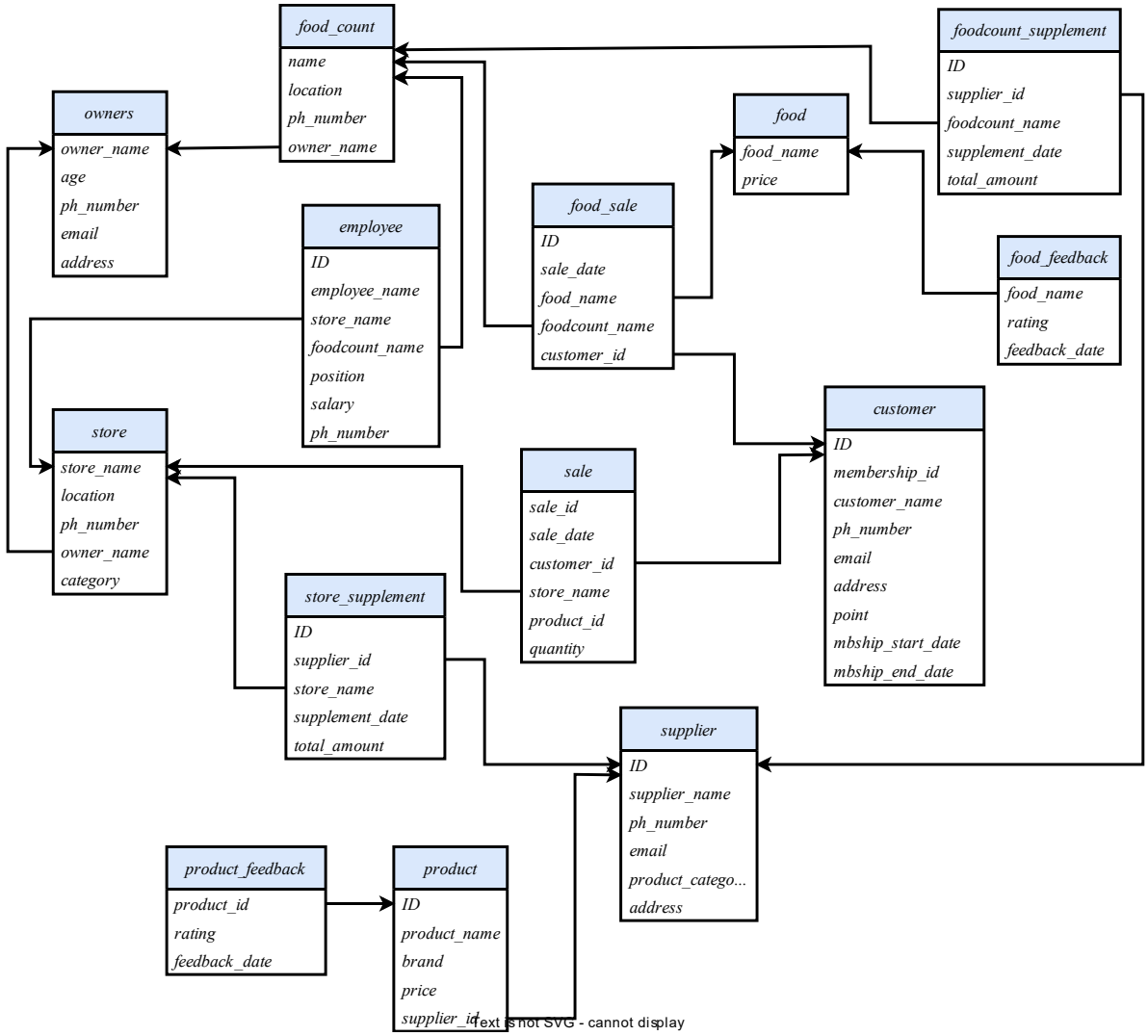


**Figure 2.1.1: Schema Diagram**

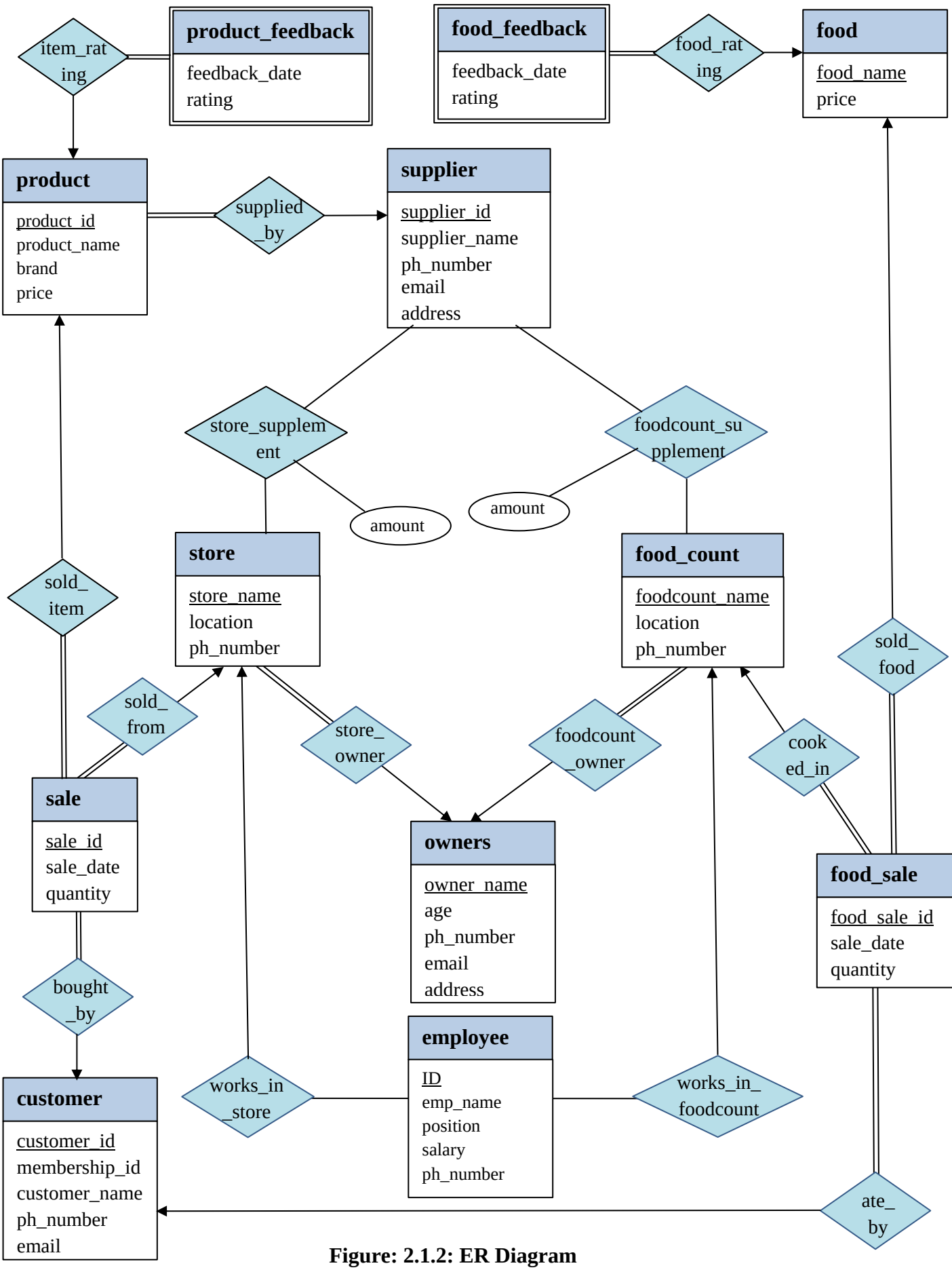


**Figure: 2.1.2: ER Diagram**

## 2.2: Data Dictionary for Shopping Mall Database

| No | Entity | Field Name | Field Type | Field Description | Nullable | Key | References |
|---|---|---|---|---|---|---|---|
| 1 | supplier | ID | bigint | unique id for supplier | No | PK | |
| | | supplier_name | varchar(60) | name of supplier | No | | |
| | | phone_number | bigint | ph no of supplier | No | | |
| | | email | varchar(60) | email of supplier | No | | |
| | | address | varchar(60) | address of supplier | No | | |
| 2 | product | product_id | bigint | unique id for product | No | PK | |
| | | product_name | varchar(60) | name of product | No | | |
| | | brand | varchar(60) | brand of product | No | | |
| | | price | decimal(9,2) | price of product | No | | |
| | | supplier_id | bigint | unique id for supplier | No | FK | supplier |
| 3 | owners | owner_name | varchar(60) | name of store owner | No | PK | |
| | | age | smallint | age of store owner | No | | |
| | | phone_number | bigint | ph no of store owner | No | | |
| | | email | varchar(60) | email of store owner | No | | |
| | | address | varchar(60) | address of store owner | No | | |
| 4 | store | store_name | varchar(60) | name of store | No | PK | |
| | | location | varchar(60) | floor that store exists | No | | |
| | | ph_number | bigint | ph no of store | No | | |
| | | owner_name | varchar(60) | name of store owner | No | FK | store |
| | | category | varchar(60) | category that store sell | No | FK | category |
| 5 | food | food_name | varchar(60) | name of food which is available in the mall | No | PK | |
| | | price | decimal(9,2) | price of the food | No | | |
| 6 | food_ count | foodcount_name | varchar(60) | name of food_count | No | PK | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | location | varchar(60) | floor that food_count exists | No | | |
| | | ph_number | bigint | ph no of food_count | No | | |
| | | owner_name | varchar(60) | name of food_count owner | No | FK | food_count |
| 7 | employee | ID | bigint | unique id for employee | No | PK | |
| | | employee_name | varchar(60) | name of employee | No | | |
| | | store_name | varchar(60) | name of store | No | FK | store |
| | | foodcount_name | varchar(60) | name of food_count | No | FK | food_count |
| | | position | varchar(60) | job position of employee | No | | |
| | | salary | decimal(9,2) | monthly salary of employee | No | | |
| | | phone_number | bigint | ph no of employee | No | | |
| 8 | customer | customer_id | bigint | unique id for customer | No | PK | |
| | | membership_id | bigint | unique id for customer with membership | Yes | | |
| | | customer_name | varchar(60) | name of customer | Yes | | |
| | | phone_number | bigint | ph no of customer | Yes | | |
| | | email | varchar(60) | email of customer | Yes | | |
| | | address | varchar(60) | address of customer | Yes | | |
| | | mbship_start_date | date | membership start date | Yes | | |
| | | mbship_end_date | date | Membership expire date | Yes | | |
| | | point | smallint | membership benefit point | Yes | | |
| 9 | sale | sale_id | bigint | unique id for sale | No | PK | |
| | | sale_date | date | date that product sold | No | | |
| | | customer_id | bigint | unique id for customer | No | FK | customer |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | store_name | varchar(60) | name of store | No | FK | store |
| | | product_id | bigint | unique id for product | No | FK | product |
| | | quantity | int | amount of product sold | No | | |
| 10 | food_sale | food_sale_id | bigint | unique id for sale of food in food_count | No | PK | |
| | | sale_date | date | date that food sold | No | | |
| | | customer_id | bigint | unique id for customer | No | FK | customer |
| | | foodcount_name | varchar(60) | name of food_count | No | FK | food_count |
| | | food_name | varchar(60) | name of food that food count sell | No | FK | food |
| | | quantity | int | amount if food sold | No | | |
| 11 | store_ supplement | supplement_id | bigint | unique id for each supplement to stores | No | PK | |
| | | supplier_id | bigint | unique id for supplier | No | FK | supplier |
| | | store | varchar(60) | name of store | No | FK | store |
| | | supplement_date | date | date that customer bought from mall | No | | |
| | | total_amount | decimal(10,2) | total amount of product sold | No | | |
| 12 | foodcount_ supplement | supplement_id | bigint | unique ID for supplements to food counts | No | PK | |
| | | supplier_id | bigint | ID of suppliers providing the supplement | No | FK | supplier |
| | | food_count | varchar(60) | name of food counts receiving the supplement | No | FK | food_count |
| | | supplement_date | date | Date that customer bought from mall | No | | |
| | | total_amount | decimal(10,2) | Total amount of food sold | No | | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 13 | product_ feedback | feedback_date | date | date that a product got a feedback | No | | |
| | | product_id | bigint | ID of product being rated | No | FK | product |
| | | rating | smallint | Rating for each product | No | | |
| 14 | food_ feedback | feedback_date | date | Date that a food got a feedback | No | | |
| | | food_name | varchar(60) | Name of food being rated | No | FK | food |
| | | ratings | smallint | Rating for each food | No | | |

**Table 2.2.1: Data Dictionary Table**

## 2.3: Relational Schema Creation using SQL

```
CREATE TABLE supplier (
    ID BIGINT AUTO_INCREMENT PRIMARY KEY,
    supplier_name VARCHAR(60) NOT NULL,
    phone_number BIGINT NOT NULL UNIQUE,
    email VARCHAR(60) NOT NULL UNIQUE,
    address VARCHAR(60) NOT NULL
)  AUTO_INCREMENT=100000;

CREATE TABLE Product (
    product_id BIGINT AUTO_INCREMENT PRIMARY KEY,
    product_name VARCHAR(60) NOT NULL,
    brand VARCHAR(60)NOT NULL,
    price NUMERIC(9 , 2 )NOT NULL,
    supplier_id BIGINT NOT NULL,
    CONSTRAINT supplied_by_whom FOREIGN KEY (supplier_id)
                REFERENCES supplier(ID)
       ON DELETE CASCADE ON UPDATE CASCADE
```

```
)AUTO_INCREMENT=20001;

CREATE TABLE owners (
   owner_name VARCHAR(60) PRIMARY KEY,
   age SMALLINT NOT NULL,
   phone_number BIGINT NOT NULL UNIQUE,
   email VARCHAR(60) NOT NULL,
   address VARCHAR(60) NOT NULL,
   CONSTRAINT check_age CHECK (age between 18 and 80)
);

CREATE TABLE store (
   store_name VARCHAR(60) PRIMARY KEY,
   location VARCHAR(60) NOT NULL,
   ph_number BIGINT NOT NULL UNIQUE,
   owner_name VARCHAR(60) NOT NULL,
   category VARCHAR(60) NOT NULL,
   CONSTRAINT store_owned_by FOREIGN KEY (owner_name)
      REFERENCES owners (owner_name)
      ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE food (
   food_name VARCHAR(60) PRIMARY KEY,
   price NUMERIC(9 , 2 ) NOT NULL
);

CREATE TABLE food_count (
   foodcount_name VARCHAR(60) PRIMARY KEY,
   location VARCHAR(60) NOT NULL,
```

```
    ph_number BIGINT NOT NULL UNIQUE,
    owner_name VARCHAR(60)NOT NULL,
    CONSTRAINT foodcount_owned_by FOREIGN KEY (owner_name)
       REFERENCES owners (owner_name)
       ON DELETE CASCADE ON UPDATE CASCADE );
CREATE TABLE employee (
    ID BIGINT AUTO_INCREMENT PRIMARY KEY,
    employee_name VARCHAR(60) NOT NULL,
    store_name VARCHAR(60) DEFAULT NULL,
    foodcount_name VARCHAR(60) DEFAULT NULL,
    position VARCHAR(60) NOT NULL,
    salary NUMERIC(9 , 2 ) NOT NULL DEFAULT 0.0,
    phone_number BIGINT NOT NULL UNIQUE,
    CONSTRAINT works_in_store FOREIGN KEY (store_name)
       REFERENCES store (store_name),
    CONSTRAINT works_in_foodcount FOREIGN KEY (foodcount_name)
       REFERENCES food_count (foodcount_name)
)  AUTO_INCREMENT=300000;

CREATE TABLE customer (
    customer_id BIGINT AUTO_INCREMENT PRIMARY KEY,
    membership_id BIGINT DEFAULT NULL,
    customer_name VARCHAR(60),
    phone_number BIGINT,
    email VARCHAR(60),
    address VARCHAR(60),
    point SMALLINT DEFAULT 0,
    mbship_start_date DATE DEFAULT '2024-1-1',
    mbship_end_date DATE DEFAULT '2024-3-31',
    CONSTRAINT check_point CHECK (point<=999),
    CONSTRAINT mbship_period_check CHECK (DATEDIFF(mbship_start_date,
mbship_end_date) <= 3)
)  AUTO_INCREMENT=400000;
```

```
CREATE TABLE sale (
   sale_id BIGINT AUTO_INCREMENT PRIMARY KEY,
   sale_date DATE NOT NULL,
   customer_id BIGINT NOT NULL,
   store_name varchar(60) NOT NULL,
   product_id BIGINT NOT NULL,
   quantity INT NOT NULL,
   CONSTRAINT bought_by FOREIGN KEY (customer_id)
      REFERENCES customer (customer_id)
                  ON DELETE CASCADE ON UPDATE CASCADE,
   CONSTRAINT bought_from_store FOREIGN KEY (store_name)
      REFERENCES store (store_name)
      ON DELETE CASCADE ON UPDATE CASCADE,
   CONSTRAINT sold_item FOREIGN KEY (product_id)
      REFERENCES product (product_id)
      ON DELETE CASCADE ON UPDATE CASCADE,
         CONSTRAINT check_countOfProduct CHECK (quantity>=1)
)AUTO_INCREMENT=100000;

CREATE TABLE food_sale (
   food_sale_id BIGINT AUTO_INCREMENT PRIMARY KEY,
   sale_date DATE NOT NULL,
   customer_id BIGINT NOT NULL,
   foodcount_name VARCHAR(60) NOT NULL,
   food_name VARCHAR(60) NOT NULL,
   quantity INT NOT NULL,
   CONSTRAINT ate_by FOREIGN KEY (customer_id)
      REFERENCES customer (customer_id)
      ON DELETE CASCADE ON UPDATE CASCADE,
   CONSTRAINT cooked_in FOREIGN KEY (foodcount_name)
      REFERENCES food_count (foodcount_name)
      ON DELETE CASCADE ON UPDATE CASCADE,
```

```
    CONSTRAINT sold_food FOREIGN KEY (food_name)
      REFERENCES food (food_name)
      ON DELETE CASCADE ON UPDATE CASCADE,
        CONSTRAINT check_countOfFood CHECK(quantity>=1)
)AUTO_INCREMENT=200000;

CREATE TABLE store_supplement (
   store_supplement_id BIGINT AUTO_INCREMENT PRIMARY KEY,
   supplier_id BIGINT NOT NULL,
   store VARCHAR(60) DEFAULT NULL,
   supplement_date DATE NOT NULL,
   total_amount NUMERIC(10 , 2 ) NOT NULL,
   CONSTRAINT product_supplement_by FOREIGN KEY (supplier_id)
      REFERENCES supplier (ID)
      ON DELETE CASCADE ON UPDATE CASCADE,
        CONSTRAINT supplement_to_store FOREIGN KEY (store)
      REFERENCES store (store_name)
      ON DELETE CASCADE ON UPDATE CASCADE
)AUTO_INCREMENT=10000;

CREATE TABLE foodcount_supplement (
   supplement_id BIGINT AUTO_INCREMENT PRIMARY KEY,
   supplier_id BIGINT NOT NULL,
   food_count VARCHAR(60) DEFAULT NULL,
   supplement_date DATE NOT NULL,
   total_amount NUMERIC(10 , 2 ) NOT NULL,
   CONSTRAINT food_supplement_by FOREIGN KEY (supplier_id)
      REFERENCES supplier (ID)
      ON DELETE CASCADE ON UPDATE CASCADE,
         CONSTRAINT supplement_to_foodcount FOREIGN KEY (food_count)
                REFERENCES food_count (foodcount_name)
      ON DELETE CASCADE ON UPDATE CASCADE
)AUTO_INCREMENT=90000;
```

```
CREATE TABLE product_feedback (
   product_id BIGINT NOT NULL,
   rating SMALLINT NOT NULL,
   feedback_date DATE NOT NULL,
   CONSTRAINT feedback_to_product FOREIGN KEY (product_id)
      REFERENCES Product (product_id)
      ON DELETE CASCADE ON UPDATE CASCADE,
         CONSTRAINT check_product_rating CHECK (rating BETWEEN 0 AND 10)
);

CREATE TABLE food_feedback (
   food_name VARCHAR(60) NOT NULL,
   rating SMALLINT NOT NULL,
   feedback_date DATE NOT NULL,
   CONSTRAINT feedback_to_food FOREIGN KEY (food_name)
      REFERENCES food (food_name)
      ON DELETE CASCADE ON UPDATE CASCADE,
         CONSTRAINT check_food_rating CHECK (rating BETWEEN 0 AND 10)
);
```

## 2.3: Data Insertion in Relational Tables

| ID | supplier_name | phone_number | email | address |
|---|---|---|---|---|
| 100000 | Khin Thet Mon | 98888888888 | ktm@gmail.com | Shwe Pyi Thar, Yangon |
| 100001 | Nov Nov Myat Noe | 93333333333 | nnmn@gmail.com | Hledan, Yangon |
| 100002 | Paing Linn Htite | 98787878787 | plh@gmail.com | Shwe Pyi Thar, Yangon |
| 100003 | Nyein Minn Tun | 99393939393 | nmt@gmail.com | Hmaw Bi, Yangon |
| 100004 | Shun Lett Htay | 94343434343 | slh@gmail.com | Mingalardon, Yangon |
| 100005 | Ye Man Aung | 95555555555 | yma@gmail.com | North Dagon, Yangon |
| 100006 | Kaung Myat Htet | 92222222222 | kmh@gmail.com | Pazundaung, Yangon |
| 100007 | Kyaw Linn Zan | 99999999999 | klz@gmail.com | Shwe Pyi Thar, Yangon |
| 100008 | Thein Htet San | 96666666666 | ths@gmail.com | Mingalardon, Yangon |
| 100009 | Hein Htet Soe | 97777777777 | hhs@gmail.com | Hledan, Yangon |
| 100012 | Pyae Phyo Kyaw | 94545454545 | ppk&gmail.com | Hledan, Yangon |

**Figure 2.3.2: Supplier table Values**

| owner_name | age | phone_number | email | address |
|---|---|---|---|---|
| Arkar Tun | 56 | 959876543233 | arkartun@gmail.com | San Chaung, Yangon |
| Aung Chan Myae | 33 | 959876543219 | maungnawmu@gmail.com | Haling Tharyar, Yangon |
| Aye Myat Noe | 45 | 959876543218 | maungkaw@gmail.com | Hledan, Yangon |
| Eae Mon Thaw | 39 | 959876543215 | eaemonthaw@gmail.com | Bo Ta Htaung, Yangon |
| Hein Minn Latt | 52 | 959876543229 | heinminnlatt@gmail.com | Pazundaung, Yangon |
| Htet Htet Yin | 37 | 959876543214 | htethtetyin@gmail.com | Mingalardon, Yangon |
| Khin Khin | 51 | 959876543228 | khinkhin@gmail.com | Myaynigone, Yangon |
| Kyaw Kyaw Win | 40 | 959876543211 | kyawkyawwin@gmail.com | Hledan, Yangon |
| Kyaw Thura | 47 | 959876543224 | kyawthura@gmail.com | Shwe Pyi Thar, Yangon |
| La Yeit Aye | 57 | 959876543234 | layeitaye@gmail.com | Myaynigone, Yangon |
| May Waddy | 54 | 959876543231 | maywaddy@gmail.com | Insein, Yangon |
| Mg Naing | 34 | 959876543220 | maungshwenaw@gmail.com | Insein, Yangon |
| Moe Thiha | 35 | 959876543210 | moethiha@gmail.com | Shwe Pyi Thar, Yangon |
| Moe Yan Myint | 48 | 959876543225 | moeyanmyintlwin@gmail.com | Haling, Yangon |
| Nanda Oo | 38 | 959876543212 | nandaoo@gmail.com | Pazundaung, Yangon |

**Figure: 2.3.3: Owners table Values**

| product_id | product_name | brand | price | supplier_id |
|---|---|---|---|---|
| 20001 | Pencil | Pilot | 500.00 | 100000 |
| 20002 | Pen | Pilot | 500.00 | 100000 |
| 20007 | Sofa | IKEA | 1497000.00 | 100002 |
| 20008 | Coffee Table | AShley | 597000.00 | 100002 |
| 20029 | Curtains | Pottery Barn | 190000.00 | 100002 |
| 20030 | Throw Pillows | Sweety Home | 25000.00 | 100002 |
| 20011 | Bread | Wonder | 2000.00 | 100003 |
| 20012 | Milk | Nestle | 2700.00 | 100003 |
| 20033 | Energy Drink | Speed | 700.00 | 100003 |
| 20034 | Biscuit | Oreo | 1200.00 | 100003 |
| 20041 | Energy Drink | Shark | 1300.00 | 100003 |
| 20013 | BASketball | Spalding | 47000.00 | 100004 |
| 20014 | Tennis Racket | Wilson | 52000.00 | 100004 |
| 20035 | Soccer Ball | Nike | 40000.00 | 100004 |
| 20036 | Golf Clubs Set | Callaway | 500000.00 | 100004 |
| 20015 | Chess Set | HASbro | 47500.00 | 100005 |
| 20016 | LEGO Set | LEGO | 25000.00 | 100005 |
| 20037 | Board Game | Monopoly | 54500.00 | 100005 |
| 20038 | Barbie Toy | Disney | 30000.00 | 100005 |
| 20023 | Notebook | Moleskine | 7600.00 | 100006 |

**Figure 2.3.4: Product table Values**

| store_name | location | phone_number | owner_name | category |
|---|---|---|---|---|
| BeatBoxz | 2nd floor | 93453453 | Su Myat | Entertainment |
| Blissful Skin | 1st floor | 9789789789 | Thiri Nanda Aung | Beauty and Personal Care |
| Bluelight Stationary | 2nd floor | 9432432432 | Eae Mon Thaw | Books and Stationery |
| Bookstore Novelties | 2nd floor | 93423423 | Moe Thiha | Books and Stationery |
| Chic & SASsy | 1st floor | 96546546 | Shwe La Thar | Apparel and FAShion |
| ClASsy Cottage | ground floor | 9908908908 | Wai Linn Tun | Home and Décor |
| Eccentric Electronics | 2nd floor | 99879879 | Nyan Linn Htet | Electronics and Technology |
| Eldora`s Collections | 2nd floor | 92342342 | Aye Myat Noe | Apparel and FAShion |
| Family Mart | 1st floor | 94534534 | Thin Yanant Phyo | Food and Beverage |
| FAShion Fizz | 1st floor | 9564564564 | Khin Khin | Apparel and FAShion |
| Glamour Evolution | 2nd floor | 93123123 | Arkar Tun | Beauty and Personal Care |

| | | | | |
|---|---|---|---|---|
| GoPlay | 1st floor | 97657657 | Sai Sai Zarni | Toys and Games |
| Health Fit | ground floor | 9098098098 | Yu War | Health and Wellness |
| Heavenly House | 1st floor | 9890890890 | Zwe Tayza | Home and Décor |
| High-Style Deco | ground floor | 9123123123 | Mg Naing | Home and Décor |
| JukeBox Junction | 1st floor | 92432432 | Kyaw Kyaw Win | Entertainment |

**Figure 2.3.5: Store table Values**

| foodcount_name | location | phone_number | owner_name |
|---|---|---|---|
| Food Cottage | 2nd floor | 9595776543227 | Phyu Phyu Win |
| Fresh Orange Chef | 2nd floor | 959376543217 | Thin Yanant Phyo |
| Iksan Korean food | 2nd floor | 959776543217 | Thin Yanant Phyo |
| Juice | 2nd floor | 959883543227 | Phyu Phyu Win |
| Lai Lai kitchen | 2nd floor | 959876543217 | Thin Yanant Phyo |
| Myeik Style | 2nd floor | 959876543317 | Thin Yanant Phyo |
| Pork & Sweet | 2nd floor | 957776543227 | Phyu Phyu Win |
| Potato Corner | 2nd floor | 959876545217 | Thin Yanant Phyo |
| Potato King | 2nd floor | 959876544727 | Phyu Phyu Win |
| Seinn Seinn Burmese food | 2nd floor | 959446543217 | Thin Yanant Phyo |
| T.G.I | 2nd floor | 959876555217 | Thin Yanant Phyo |
| Thai Ser | 2nd floor | 959866543227 | Phyu Phyu Win |
| Thar Li Swa | 2nd floor | 959875543227 | Phyu Phyu Win |
| The Spicy House | 2nd floor | 958576543227 | Phyu Phyu Win |
| The Tree Thai | 2nd floor | 959895543227 | Phyu Phyu Win |
| V-Hangout | 2nd floor | 959876592227 | Phyu Phyu Win |

**Figure 2.3.6: Food_Count table Values**

+

| food_name | price |
|---|---|
| BBQ Ribs | 6500.00 |
| Beef Stroganoff | 5500.00 |
| Burger | 3000.00 |
| Burrito | 2500.00 |
| Calamri | 3550.00 |
| CarnitAS | 4000.00 |
| Cesar Salad | 3000.00 |
| Ceviche | 3000.00 |
| Chicken Caesar Salad | 1800.00 |
| Chicken Wings | 4500.00 |
| Chili | 2000.00 |
| Cobb Salad | 4000.00 |
| Crab Cakes | 6500.00 |
| Egg Rolls | 1500.00 |
| Eggplant Parmesan | 4500.00 |
| FajitAS | 3500.00 |
| Falafel | 1200.00 |
| Fish and Chips | 5000.00 |
| Fish Tacos | 4000.00 |
| Fried Rice | 2000.00 |
| Grilled Salmon | 8000.00 |
| Gyros | 2500.00 |
| Hamburger | 2000.00 |
| Hot Dog | 1500.00 |
| LASagna | 5500.00 |

**Figure 2.3.7: Food table Values**

| ID | employee_name | store_name | foodcount_name | position | salary | phone_number |
|---|---|---|---|---|---|---|
| 11003 | Min Khant Si Thu | PerformActive | | Store Manager | 400000.00 | 9123789456 |
| 11006 | Tin Than Moe | Marvel Toys | | Toy Specialist | 400000.00 | 9142678456 |
| 11007 | San San Tin | The Bookshelf | | Book Seller | 250000.00 | 9142666456 |
| 11014 | A Nadi Phyo | Chic & SASsy | | Brand Manager | 450000.00 | 9123456999 |
| 11015 | Ei Mon Zin | Urban Elegance | | Sales Manager | 400000.00 | 9123456677 |
| 11017 | Htun Yadanar Oo | Health Fit | | Sales Manager | 300000.00 | 9123456679 |
| 11023 | Eaint Hmue Thwel | Skin Artistry | | Facial Specialist | 350000.00 | 9123456777 |
| 11025 | Zaw Ye Htet | ClASsy Cottage | | CAShier | 250000.00 | 9123458787 |
| 11026 | Hsu Pyae Yati | StyleStanza | | Store Manager | 450000.00 | 9123458788 |
| 11027 | Phone Min Khant | JukeBox Junction | | Manager | 400000.00 | 9123458778 |
| 11031 | Hsu Yadanar | Skin Artistry | | Makeup Artist | 350000.00 | 9123456688 |
| 11034 | Po Po | BeatBoxz | | CAShier | 250000.00 | 9123678495 |
| 11035 | Kyawt Kyawt | The Bookshelf | | Book Seller | 250000.00 | 9123678478 |
| 11038 | Chit Po | Health Fit | | CAShier | 250000.00 | 9123957984 |
| 11040 | Ko Thu | ClASsy Cottage | | CAShier | 250000.00 | 9126557498 |
| 11042 | May Thu | | Big Bowl | Chef | 500000.00 | 9123876945 |
| 11043 | Maung Maung | | D'PENYETZ | Chef | 500000.00 | 9123765849 |
| 11045 | La Pyae | | Potato King | CAShier | 250000.00 | 9123675894 |

**Figure 2.3.8: Employee table Values**

| customer_id | membership_ id | customer_ name | phone_number | email | address | point | mbship_start_date | mbship_end_date |
|---|---|---|---|---|---|---|---|---|
| 400000 | | Swan | 9765349821 | swam@gmail.com | Mayangone, Yangon | 0 | 2024-01-04 | 2024-04-04 |
| 400001 | | Dar Dar | 9798543230 | dardar@gmail.com | Mayangone, Yangon | 0 | 2024-01-01 | 2024-04-01 |
| 400002 | | Yu Ri | 9778903334 | yuri@gmail.com | Yan Kin,Yangon | 0 | 2024-01-01 | 2024-04-01 |
| 400005 | | Hlan | 9788776669 | hlan@gmail.com | Kamayut,Yangon | 0 | 2024-01-03 | 2024-04-03 |
| 400008 | | Myo Win | 9734587655 | myowin@gmail.com | Pazuntaung, Yangon | 0 | 2024-01-05 | 2024-04-05 |
| 400009 | | Aung Phone | 9278097644 | aungphone@gmail.com | Su Lay,Yangon | 0 | 2024-01-05 | 2024-04-05 |
| 400010 | | Juu | 9709878900 | juu@gmail.com | Su Lay | 0 | 2024-01-06 | 2024-04-06 |
| 400011 | | Ei Phue | 9749020902 | eiphue@gmail.com | Shwe Pyi Thar,Yangon | 0 | 2024-01-06 | 2024-04-06 |
| 400012 | | Lamin | 9776568900 | lamin@gmail.com | Hledan,Yangon | 0 | 2024-01-07 | 2024-04-07 |
| 400013 | | Bell | 9767896577 | bell@gmail.com | Tarmwe,Yangon | 0 | 2024-01-07 | 2024-04-07 |
| 400014 | | Yamin | 9765890622 | yamin@gmail.com | Tamine,Yangon | 0 | 2024-01-08 | 2024-04-08 |
| 400015 | | Soe Myint | 9709567122 | soemyint@gmail.com | Kamayut,Yangon | 0 | 2024-01-08 | 2024-04-08 |
| 400016 | | Htal | 9728964733 | htal@gmail.com | Kamayut,Yangon | 0 | 2024-01-09 | 2024-04-09 |
| 400017 | 2000100 | Yu Ji | 9872345455 | yuji55@gmail.com | Insein,Yangon | 250 | 2024-01-09 | 2024-04-09 |

**Figure 2.3.9: Customer table Values**

| sale_id | sale_date | customer_id | store_name | product_id | quantity |
|---|---|---|---|---|---|
| 100000 | 2024-01-01 | 400001 | Eldora`s Collections | 20003 | 2 |
| 100001 | 2024-01-01 | 400002 | Chic & SASsy | 20004 | 3 |
| 100002 | 2024-01-02 | 400003 | Urban Elegance | 20025 | 2 |
| 100003 | 2024-01-02 | 400004 | FAShion Fizz | 20026 | 1 |
| 100004 | 2024-01-03 | 400005 | Eldora`s Collections | 20003 | 3 |
| 100005 | 2024-01-03 | 400006 | Blissful Skin | 20009 | 2 |
| 100006 | 2024-01-04 | 400007 | Glamour Evolution | 20010 | 1 |
| 100007 | 2024-01-04 | 400000 | Youthful Elixir | 20021 | 3 |
| 100008 | 2024-01-05 | 400008 | Skin Artistry | 20022 | 2 |
| 100009 | 2024-01-05 | 400009 | Blissful Skin | 20031 | 3 |
| 100010 | 2024-01-06 | 400010 | Glamour Evolution | 20032 | 2 |
| 100011 | 2024-01-06 | 400011 | The Bookshelf | 20001 | 10 |
| 100012 | 2024-01-07 | 400012 | Bluelight Stationary | 20002 | 10 |
| 100013 | 2024-01-07 | 400013 | Bookstore Novelties | 20023 | 5 |
| 100014 | 2024-01-08 | 400014 | The Bookshelf | 20024 | 2 |
| 100015 | 2024-01-08 | 400015 | Techeon | 20005 | 1 |
| 100016 | 2024-01-09 | 400016 | Techeon | 20006 | 1 |
| 100017 | 2024-01-09 | 400017 | Eccentric Electronics | 20027 | 3 |
| 100018 | 2024-01-10 | 400018 | Eccentric Electronics | 20028 | 3 |
| 100019 | 2024-01-10 | 400019 | BeatBoxz | 20019 | 3 |

**Figure 2.3.10: Sale table Values**

| food_sale_id | sale_date | customer_id | foodcount_name | food_name | quantity |
|---|---|---|---|---|---|
| 200000 | 2024-03-01 | 400000 | Potato Corner | Pizza | 5 |
| 200001 | 2024-03-02 | 400001 | Potato Corner | Burger | 6 |
| 200002 | 2024-03-02 | 400002 | Lai Lai kitchen | Salad | 10 |
| 200003 | 2024-03-04 | 400003 | Lai Lai kitchen | PASta | 5 |
| 200004 | 2024-03-05 | 400004 | Big Bowl | Steak | 3 |
| 200005 | 2024-03-06 | 400005 | Fresh Orange Chef | Sushi | 8 |
| 200006 | 2024-03-06 | 400006 | Big Bowl | Chicken Wings | 10 |
| 200007 | 2024-03-10 | 400007 | D'PENYETZ | Tacos | 10 |
| 200008 | 2024-03-10 | 400008 | Fresh Orange Chef | Fried Rice | 5 |
| 200009 | 2024-03-10 | 400009 | D'PENYETZ | Sandwich | 5 |
| 200010 | 2024-03-11 | 400010 | Du Du Mao Cia | Burrito | 4 |
| 200011 | 2024-03-12 | 400011 | Du Du Mao Cia | Soup | 9 |
| 200012 | 2024-03-15 | 400012 | Fresh Orange Chef | Fish and Chips | 8 |
| 200013 | 2024-03-15 | 400013 | Iksan Korean food | LASagna | 6 |
| 200014 | 2024-03-16 | 400014 | Iksan Korean food | Sushi Roll | 10 |
| 200015 | 2024-03-16 | 400015 | Myeik Style | Hot Dog | 10 |
| 200016 | 2024-03-17 | 400016 | Myeik Style | FajitAS | 6 |
| 200017 | 2024-03-19 | 400017 | Seinn Seinn Burmese Food | Ramen | 20 |

**Figure 2.3.11: Food_sale table Values**

| product_id | rating | feedback_date |
|---|---|---|
| 20002 | 3 | 2024-01-13 |
| 20003 | 4 | 2024-02-21 |
| 20004 | 5 | 2024-01-10 |
| 20005 | 5 | 2024-01-23 |
| 20006 | 3 | 2024-01-19 |
| 20007 | 4 | 2024-01-17 |
| 20008 | 4 | 2024-01-14 |
| 20009 | 4 | 2024-01-13 |
| 20010 | 4 | 2024-01-01 |
| 20011 | 4 | 2024-01-03 |
| 20012 | 5 | 2024-01-03 |
| 20013 | 4 | 2024-01-05 |
| 20014 | 3 | 2024-01-07 |
| 20015 | 4 | 2024-01-04 |
| 20016 | 4 | 2024-01-29 |
| 20017 | 5 | 2024-01-22 |
| 20018 | 4 | 2024-01-26 |
| 20019 | 3 | 2024-01-24 |
| 20020 | 5 | 2024-01-27 |

**Figure 2.3.12: Product_feedback table Values**

| food_name | rating | feedback_date |
|---|---|---|
| BBQ Ribs | 4 | 2024-01-23 |
| Beef Stroganoff | 3 | 2024-01-13 |
| Burger | 4 | 2024-02-21 |
| Burrito | 5 | 2024-01-10 |
| Calamri | 5 | 2024-01-23 |
| CarnitAS | 3 | 2024-01-19 |
| Cesar Salad | 4 | 2024-01-17 |
| Cesar Salad | 4 | 2024-01-14 |
| Ceviche | 4 | 2024-01-13 |
| Chicken Caesar Salad | 4 | 2024-01-01 |
| Chicken Wings | 4 | 2024-01-03 |
| Chili | 5 | 2024-01-03 |
| Cobb Salad | 4 | 2024-01-05 |
| Crab Cakes | 3 | 2024-01-07 |
| Egg Rolls | 4 | 2024-01-04 |
| Eggplant Parmesan | 4 | 2024-01-29 |
| FajitAS | 5 | 2024-01-22 |
| Falafel | 4 | 2024-01-26 |
| Fish and Chips | 3 | 2024-01-24 |

**Figure 2.3.13: Food_feedback table Values**

| supplement_id | supplier_id | store | supplement_date | total_amount |
|---|---|---|---|---|
| 10000 | 100000 | High-Style Deco | 2024-02-13 | 150000.00 |
| 10001 | 100001 | Eldora`s Collections | 2024-02-14 | 160000.00 |
| 10002 | 100002 | BeatBoxz | 2024-02-15 | 170000.00 |
| 10003 | 100003 | PerformActive | 2024-02-16 | 180000.00 |
| 10004 | 100004 | Pure-life Wellness | 2024-02-17 | 190000.00 |
| 10005 | 100005 | Techeon | 2024-02-18 | 200000.00 |
| 10006 | 100006 | Marvel Toys | 2024-02-19 | 210000.00 |
| 10007 | 100007 | The Bookshelf | 2024-02-20 | 220000.00 |
| 10008 | 100008 | Blissful Skin | 2024-02-21 | 230000.00 |
| 10009 | 100009 | Heavenly House | 2024-02-22 | 240000.00 |
| 10010 | 100010 | Glamour Evolution | 2024-02-23 | 250000.00 |
| 10011 | 100011 | Bluelight Stationary | 2024-02-24 | 260000.00 |
| 10012 | 100012 | Local Color Liquor | 2024-02-25 | 270000.00 |
| 10013 | 100012 | Chic & SASsy | 2024-02-26 | 280000.00 |
| 10014 | 100011 | GoPlay | 2024-02-27 | 290000.00 |
| 10015 | 100010 | Urban Elegance | 2024-02-28 | 300000.00 |
| 10016 | 100009 | Eccentric Electronics | 2024-02-29 | 310000.00 |
| 10017 | 100008 | Health Fit | 2024-01-01 | 320000.00 |
| 10018 | 100007 | Youthful Elixir | 2024-01-02 | 330000.00 |
| 10019 | 100006 | Bookstore Novelties | 2024-01-03 | 340000.00 |

**Figure 2.3.14: Store_supplement table Values**

| supplement_id | supplier_id | food_count | supplement_date | total_amount |
|---|---|---|---|---|
| 90000 | 100003 | Arr Mei | 2024-01-05 | 360000.00 |
| 90001 | 100004 | Big Bowl | 2024-01-06 | 370000.00 |
| 90002 | 100005 | D'PENYETZ | 2024-01-07 | 380000.00 |
| 90003 | 100006 | Du Du Mao Cia | 2024-01-08 | 390000.00 |
| 90004 | 100007 | Food Cottage | 2024-01-09 | 400000.00 |
| 90005 | 100008 | Fresh Orange Chef | 2024-01-10 | 410000.00 |

| | | | | |
|---|---|---|---|---|
| 90006 | 100009 | Iksan Korean food | 2024-01-11 | 420000.00 |
| 90007 | 100010 | Lai Lai kitchen | 2024-01-12 | 430000.00 |
| 90008 | 100011 | Myeik Style | 2024-01-13 | 440000.00 |
| 90009 | 100012 | Pork & Sweet | 2024-01-14 | 450000.00 |
| 90010 | 100011 | Potato Corner | 2024-01-15 | 460000.00 |
| 90011 | 100010 | Potato King | 2024-01-16 | 470000.00 |
| 90012 | 100009 | Seinn Seinn Burmese food | 2024-01-17 | 480000.00 |
| 90013 | 100008 | T.G.I | 2024-01-18 | 490000.00 |
| 90014 | 100007 | Thai Ser | 2024-01-19 | 500000.00 |
| 90015 | 100006 | Thar Li Swa | 2024-01-20 | 510000.00 |
| 90016 | 100005 | The Spicy House | 2024-01-21 | 520000.00 |
| 90017 | 100004 | The Tree Thai | 2024-01-22 | 530000.00 |
| 90018 | 100003 | V-Hangout | 2024-01-23 | 540000.00 |

**Figure 2.3.15: food_count_supplement table Values**

## 2.4: Customer Requirements

1. Find all the store names which are under the category of “Beauty and Personal Care”.
2. Find the food names of the top three highest rated foods respectively.
3. Find the product ID, product name, brand name and price of every product supplied by “Thein Htet San”.
4. Update the 'customer' table based on purchases recorded in the 'sale' table. If a customer spends over certain amounts on a single item, they earn membership points automatically. If a customer spends over 5,000,000, they get 999 points. If they spend over 2,000,000, they get 750 points. If over 1,000,000, they get 500 points, and if over 500,000, they get 250 points. However, if a customer already has more than 250 points and spends over 500,000, their points stay the same. Also, set the membership start date to the sale date and the membership end date to three months after the sale date. Update the 'customer' table accordingly.
5. Update the 'customer' table by resetting the membership start date, membership end date, and membership points to null for customers whose membership end date has already passed. Ensure that the update is executed whenever the membership end date for a customer is in the past.
6. Create two views: 'sale_description' and 'food_sale_description'. 'sale_description' should display detailed sales information including product price, total sale amount, calculated discount based on customer points, and final cost after discount. Ensure results are ordered by cost in descending order. 'food_sale_description' should display similar information for food sales, also ordered by cost for clarity.

7. Using the views 'sale_description' and 'food_sale_description', retrieve the top three stores and food counts based on total sales within specific intervals of months.
8. Find the top three best-selling products and foods respectively within a particular month interval, considering total quantities sold.
9. Find the top three best-selling days within a particular month interval in terms of total cost of items sold.
10. Find the profit of each store in a particular month by subtracting the total sales with the total supplement amount in that month. Find the profit of each food count as well with the same process.

## 2.5: Query Planning with Algebra

1. Find all the store names which are under the category of "Beauty and Personal Care".

   $\Pi_{\text{store\_name}}(\sigma_{\text{category= 'Beauty and Personal Care'}}(\text{store}))$

2. Find the average rating of products and foods along with the respective product names and food names.

   (i) Average Rating of each food

   $\gamma_{\text{food\_name,avg(rating)}\rightarrow\text{feedback}}(\text{food\_feedback})$

   (ii) Top five highest rated product names

   $\gamma_{\text{product\_name,avg(rating)}\rightarrow\text{feedback}}$

   $(\text{product\_feedback} \bowtie_{\text{product\_feedback.product\_id = product.product\_id}} \text{product})$

3. Find the product ID, product name, brand name and price of every product supplied by "Thein Htet San".

   $\Pi_{\text{product\_id, product\_name, supplier\_name}}(\sigma_{\text{supplier\_name='Thein Htet San'}}$

   $(\text{product} \bowtie_{\text{product.supplier\_id = supplier.ID}} \text{supplier}))$

4. Create two views: 'sale_description' and 'food_sale_description'. 'sale_description' should display detailed sales information including product price, total sale amount, calculated discount based on customer points, and final cost after discount. Ensure results are ordered by cost in descending order. 'food_sale_description' should display similar information for food sales, also ordered by cost for clarity.

(i) Creating 'sale_description' view

$\Pi_{s.*,\ p.price,\ (s.quantity*p.price)\rightarrow total,\ c.point/100\rightarrow discount,\ (s.quantity*p.price-s.quantity*p.price*c.point/10000)\rightarrow cost}\ (\rho_{p,s,c}\ (product \bowtie_{s.product\_id=p.product\_id} sale \bowtie_{s.customer\_id=c.customer\_id} customer))$

(ii) Creating 'food_sale_description' view

$\Pi_{s.*,\ f.price,\ (s.quantity*f.price)\rightarrow total,\ c.point/100\rightarrow discount,\ (s.quantity*f.price-s.quantity*f.price*c.point/10000)\rightarrow cost}\ (\rho_{f,s,c}\ (food \bowtie_{s.product\_id=p.product\_id} food\_sale \bowtie_{s.customer\_id=c.customer\_id} customer))$

5. Find the top five best-selling products and foods respectively between January and April, considering total quantities sold.

(i) Top five best-selling products between January and April

$\lambda_5(\sigma_{order\ by\ Quantity\ desc}(\Pi_{sale\_description.product\_id,\ Product.product\_name,\ Quantity}(\gamma_{product\_id,SUM(quantity)\rightarrow Quantity}(\sigma_{MONTH(sale\_date)\ BETWEEN\ 1\ AND\ 4}(sale\_description \bowtie_{sale\_description.product\_id\ =\ product.product\_id} product)))))$

(ii) Top five best-selling foods between January and April

$\lambda_5(\sigma_{order\ by\ Quantity\ DESC}(\Pi_{food\_name,\ Quantity}(\gamma_{food\_name,SUM(quantity)\rightarrow Quantity}(\sigma_{MONTH(sale\_date)\ BETWEEN\ 1\ AND\ 4}(food\_sale\_description)))))$

## 2.6: Query Execution with MySQL

1. Find all the store names which are under the category of “Beauty and Personal Care”.

   SELECT store_name FROM store WHERE category=”Beauty and Personal Care”;

| store_name |
|---|
| Blissful Skin |
| Glamour Evolution |
| Skin Artistry |
| Youthful Elixir |

**Fig: Query of stores under the “Beauty and Personal Care” category**

2. Find the food names of the top three highest rated foods respectively.

   SELECT food_name, AVG(rating) FROM food_feedback GROUP BY food_name ORDER BY rating DESC LIMIT 3;

| food_name | rating |
|---|---|
| BBQ Ribs | 4.0000 |
| Beef Stroganoff | 3.0000 |
| Burger | 4.0000 |

**Figure 2.6.2: Query of top three highest rated foods**

3. Find the product ID, product name, brand name and price of every product supplied by “Thein Htet San”.
   SELECT product_id, product_name, brand FROM product JOIN supplier ON supplier.ID=product.supplier_id WHERE supplier_name="Thein Htet San";

| product_id | product_name | brand |
|---|---|---|
| 20019 | Movie DVD | Warner Bros |
| 20020 | Video Game | Sony |

**Figure 2.6.3: Query of stores that Thein Htet San supplies to**

4. Update the 'customer' table based on purchases recorded in the 'sale' table. If a customer spends over certain amounts on a single item, they earn membership points automatically. If a customer spends over 5,000,000, they get 999 points. If they spend over 2,000,000, they get 750 points. If over 1,000,000, they get 500 points, and if over 500,000, they get 250 points. However, in case a customer already has more than 250

points and spends over 500,000, their points stay the same. Also, set the membership start date to the sale date and the membership end date to three months after the sale date. Update the 'customer' table accordingly.

```
UPDATE customer AS c
INNER JOIN sale AS s USING(customer_id)
LEFT JOIN product AS p USING(product_id)
SET c.point = CASE
      WHEN p.price * s.quantity > 500000 AND c.point<250 THEN 250
      WHEN p.price * s.quantity > 1000000 AND c.point<500 THEN 500
      WHEN p.price * s.quantity > 2000000 AND c.point<750 THEN 750
      WHEN p.price * s.quantity > 5000000 AND c.point<999 THEN 999
      ELSE c.point
         END,
   c.mbship_start_date = s.sale_date,
   c.mbship_end_date = DATE_ADD(s.sale_date, INTERVAL 3 MONTH)
WHERE c.customer_id IS NOT NULL;
```

| customer_id | membership_ id | customer_ name | phone_number | email | address | point | mbship_start_date | mbship_end_date |
|---|---|---|---|---|---|---|---|---|
| 400015 | | Soe Myint | 970956712 | soemyint@gmail.com | Kamayut,Yangon | 500 | 2024-01-08 | 2024-04-08 |
| 400016 | | Htal | 972896473 | htal@gmail.com | Kamayut,Yangon | 750 | 2024-01-09 | 2024-04-09 |

**Figure 2.6.4: Modified membership point values after updating the membership points according to the purchase amount**

5. Update the 'customer' table by resetting the membership start date, membership end date, and membership points to null for customers whose membership end date has already passed. Ensure that the update is executed whenever the membership end date for a customer is in the past.

```
UPDATE customer SET point=0, mbship_end_date=NULL WHERE
customer.mbship_end_date<CURRENT_DATE;
```

6. Create two views: 'sale_description' and 'food_sale_description'. 'sale_description' should display detailed sales information including product price, total sale amount, calculated discount based on customer points, and final cost after discount. Ensure results are ordered by cost in descending order. 'food_sale_description' should display similar information for food sales, also ordered by cost for clarity.

```
CREATE VIEW sale_description AS
(
        SELECT s.*, p.price, (s.quantity*p.price) AS total, c.point/100 AS discount,
                (s.quantity*p.price-s.quantity*p.price*c.point/10000) AS cost
        FROM sale AS s JOIN product AS p USING(product_id) JOIN customer AS
c USING(customer_id) ORDER BY cost DESC
);
CREATE VIEW food_sale_description AS
(
        SELECT food_sale.*, price, (price*quantity) AS total,(point/100) AS
discount, (price*quantity-price*quantity*point/10000) AS cost
    FROM food_sale JOIN food USING (food_name) JOIN customer
USING(customer_id) ORDER BY cost
);
```

| sale_id | sale_date | customer_id | store_name | product_id | quantity | price | total | discount | cost |
|---|---|---|---|---|---|---|---|---|---|
| **100030** | 2024-01-16 | 400030 | High-Style Deco | 20007 | 2 | 1497000.00 | 2994000.00 | 7.5000 | 2769450.000000 |
| **100060** | 2024-01-31 | 400023 | Techeon | 20006 | 1 | 2345000.00 | 2345000.00 | 0.3400 | 2337027.000000 |
| **100016** | 2024-01-09 | 400016 | Techeon | 20006 | 1 | 2345000.00 | 2345000.00 | 7.5000 | 2286375.000000 |
| **100082** | 2024-02-11 | 400036 | VIAAL Active Wear | 20036 | 4 | 500000.00 | 2000000.00 | 0.6500 | 1987000.000000 |
| **100037** | 2024-01-19 | 400037 | VIAAL Active Wear | 20036 | 4 | 500000.00 | 2000000.00 | 5.0000 | 1900000.000000 |

**Figure 2.6.5: sale_description view values**

| food_sale_id | sale_date | customer_id | foodcount_name | food_name | quanti ty | price | total | discount | cost |
|---|---|---|---|---|---|---|---|---|---|
| 200036 | 2024-04-07 | 400035 | Potato King | Vegetable Soup | 3 | 3000.00 | 9000.00 | 1.7400 | 8843.400000 |
| 200042 | 2024-04-16 | 400041 | Potato King | Pulled Pork Sandwich | 5 | 1800.00 | 9000.00 | 0.5400 | 8951.400000 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 200045 | 2024-04-18 | 400044 | The Tree Thai | Miso Soup | 6 | 1500.00 | 9000.00 | 0.3000 | 8973.000000 |
| 200008 | 2024-03-10 | 400008 | Fresh Orange Chef | Fried Rice | 5 | 2000.00 | 10000.00 | 0.0000 | 10000.000000 |
| 200010 | 2024-03-11 | 400010 | Du Du Mao Cia | Burrito | 4 | 2500.00 | 10000.00 | 0.0000 | 10000.000000 |

**Figure 2.6.6: food_sale_description view values**

7. Using the views 'sale_description' and 'food_sale_description', retrieve the top three stores and food counts based on total sales within specific intervals of months.

```
SELECT store_name, SUM(cost) AS total_sales FROM sale_description
        WHERE sale_date BETWEEN '2023-1-1' AND '2024-2-1'
        GROUP BY store_name ORDER BY total_sales DESC LIMIT 3;
SELECT foodcount_name, SUM(cost) AS total_sales
        FROM food_sale_description
                WHERE sale_date BETWEEN '2023-1-1' AND '2024-7-1'
                GROUP BY foodcount_name ORDER BY total_sales DESC LIMIT
                3;
```

| store_name | total_sales |
|---|---|
| Techeon | 6719402.000000 |
| Eccentric Electronics | 3177369.000000 |
| High-Style Deco | 2893125.000000 |

**Figure 2.6.7: Query for top three best-selling stores**

| foodcount_name | total_sales |
|---|---|
| Food Cottage | 513026.400000 \| |
| Juice | 480593.800000 |
| T.G.I | 421812.900000 |

**Figure:2.6.8: Query for top three best-selling food counts**

8. Find the top three best-selling products within a particular month interval, considering total quantities sold.

```
SELECT sale_description.product_id, product.product_name,
        SUM(quantity) AS Quantity
FROM sale_description JOIN product using (product_id)
        WHERE MONTH(sale_date) BETWEEN 1 AND 4 GROUP BY product_id
        ORDER BY Quantity DESC LIMIT 3;
```

| product_id | product_name | quantity |
|---|---|---|
| 20001 | Pencil | 30 |
| 20002 | Pen | 30 |
| 20003 | T-Shirt | 17 |

**Figure 2.6.9: Query for top three best seller products**

9. Find the top three best-selling days within a particular month interval in terms of total cost of items sold.

```
SELECT sale_date, SUM(cost) AS total_sales
FROM sale_description
    WHERE MONTH(sale_date)=1 AND YEAR(sale_date)=2024
    GROUP BY sale_date ORDER BY total_sales DESC LIMIT 3;
```

| sale_date | total_sales |
|---|---|
| 2024-01-16 | 4470900.000000 |
| 2024-01-31 | 3823527.000000 |
| 2024-01-09 | 3748875.000000 |

**Figure 2.6.10: Query for top three best selling days**

10. Find the profit of each store from January to March by subtracting the total sales with the total supplement amount in that interval. Find the profit of each food count as well with the same process. Sort the result by the profit amount in a descending order.

```
SELECT
sale_description.store_name,
SUM(sale_description.cost) AS total_sales,
SUM(store_supplement.total_amount) AS total_store_supplement,
SUM(sale_description.cost) - SUM(store_supplement.total_amount) AS profit
    FROM sale_description JOIN store_supplement
        ON sale_description.store_name = store_supplement.store
    WHERE (MONTH(store_supplement.supplement_date) BETWEEN 1 AND
    3)  AND (MONTH(sale_description.sale_date) BETWEEN 1 AND 3)
    GROUP BY store_name
    ORDER BY profit DESC;
```

| store_name | total_sales | total_store_supplement | profit |
|---|---|---|---|
| Techeon | 6719402.000000 | 800000.00 | 5919402.000000 |
| Heavenly House | 3402900.000000 | 480000.00 | 2922900.000000 |
| VIAAL Active Wear | 1480000.000000 | 1480000.00 | 2711808.400000 |
| High-Style Deco | 3017725.000000 | 450000.00 | 2567725.000000 |
| Eccentric Electronics | 3177369.000000 | 1240000.00 | 1937369.000000 |
| ClASsy Cottage | 2210916.000000 | 800000.00 | 1410916.000000 |
| Chic & SASsy | 574521.600000 | 840000.00 | -265478.400000 |
| BeatBoxz | 35406.000000 | 340000.00 | -304594.000000 |
| Urban Elegance | 561574.800000 | 900000.00 | -338425.200000 |
| Eldora`s Collections | 758169.000000 | 1120000.00 | -361831.000000 |
| PerformActive | 343931.200000 | 720000.00 | -376068.800000 |
| GoPlay | 148005.000000 | 580000.00 | -431995.000000 |
| PlayKid | 320803.350000 | 780000.00 | -459196.650000 |
| Pure-life Wellness | 358646.400000 | 950000.00 | -591353.600000 |

**Figure 2.6.11: Query for finding profit of each store from January to March**

# Chapter 3

# Conclusion

The development and implementation of the shopping mall database management system (DBMS) represents a significant advancement in the organization and management of retail operations within shopping malls. This project actively contributes to the digitalization of shopping mall sales processes, offering a comprehensive set of functionalities that benefit various stakeholders.

Through the structured organization of inventory, sales, and customer data, the system facilitates the streamlining and automation of key tasks performed by mall staff, leading to enhanced operational efficiency. Additionally, the integration of a customer rating system empowers mall management to identify areas for improvement within individual stores and food establishments, enabling the implementation of data-driven strategies to elevate customer satisfaction and product quality. Furthermore, the system automates critical calculations associated with total sales, profits, and staff salaries, providing both mall management and individual tenants with valuable real-time insights into their performance. This comprehensive data analysis fosters informed decision-making, empowering stakeholders to optimize resource allocation, pricing strategies, and inventory management. By embracing this novel approach to data management, shopping malls are positioned to not only navigate the ever-evolving retail landscape but also thrive within it. In conclusion, the shopping mall database management system stands as a testament to the transformative potential of leveraging technology to optimize operations and enhance the overall experience for all stakeholders – mall managers, tenants, and ultimately, the customers themselves.