

CPSC 304 Project Cover Page

Milestone #: 2

Date: 02/08 2024

Group Number: 53

| Name | Student Number | CS Alias (Userid) | Preferred E-mail Address |
|------------|----------------|-------------------|---------------------------|
| Ernest Gao | 93408227 | b5w6x | ernestgao2021@outlook.com |
| Roger Qi | 98746993 | c7t3u | qilinghan2020@126.com |
| Ryan Ge | 78444452 | o3f7b | sunedujunmingge@yahoo.com |

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

A brief project description:

This program is aimed to form a database that is used for food searching and health problems. The domain of the project lies in the study of food science specifically focusing on food health and food safety related to humans. The program here can identify what kind of nutrients this food has and what food services can provide this food. In that case this database can be used to justify the nutritional value of food on the menu and give suggestions on some specific requirements for what is going to be eaten, like suggestions of restaurants or grocery stores.

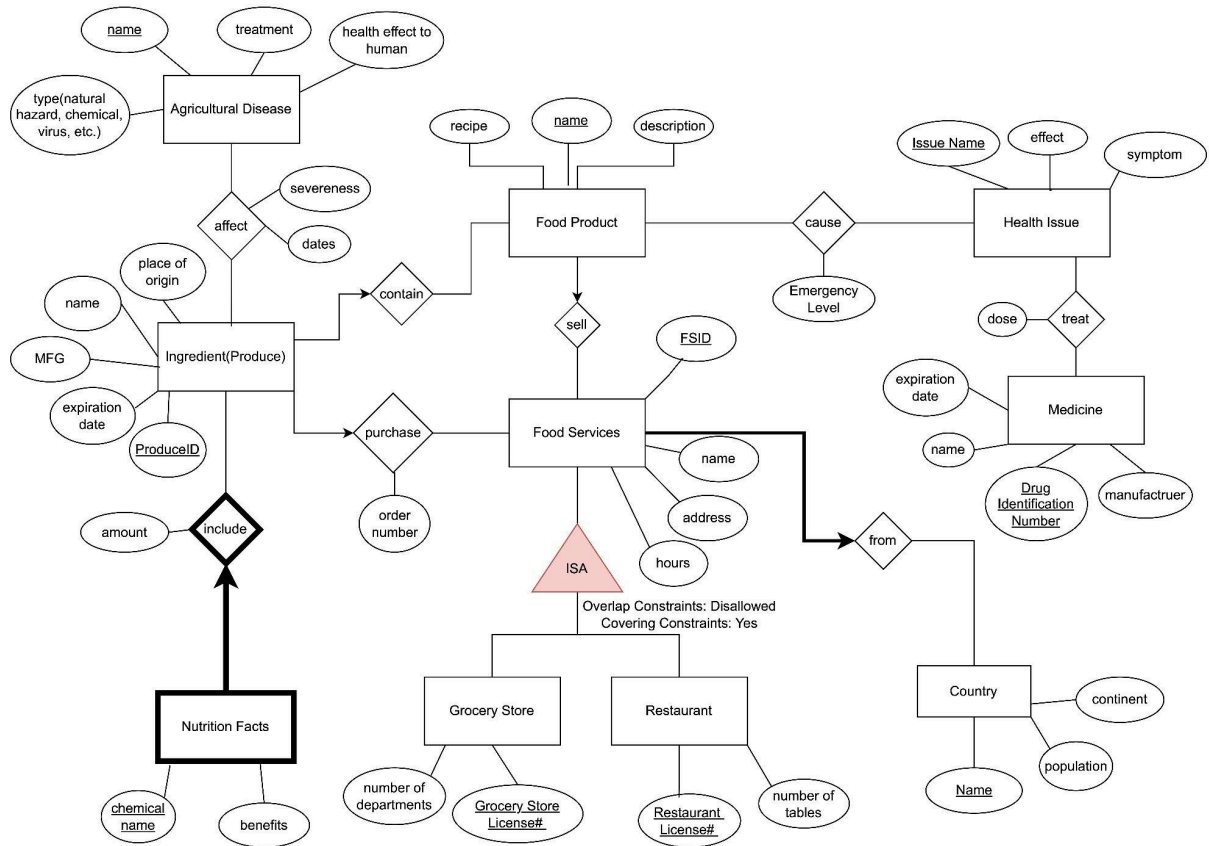
Database specifications:

The database will provide users with the function of searching and retrieving all kinds of food information, including its nutritional content and potential health benefits or concerns. Users will also be able to find food service organizations such as restaurants and grocery stores, and filter according to the types of dishes and the standards of food and accommodation. In addition, the database can provide recommendation functions according to users' preferences, dietary restrictions or health goals, so as to help users make wise choices about their food consumption and obtain appropriate food services.

Description of the application platform:

We will use the department provided Oracle for our database. For our technology stack, we will use PHP to implement the application.

ER-Diagram:



Schema:

FoodProductSell(FPName: char(20), Description: varchar, Recipe: varchar, **FSID**: int)
PK: (FPName)
FK: FSID references FoodServiceFrom

FoodServicesFrom(FSID: int, FSName: char(20), FSAddress: varchar, hours: varchar, **CountryName**: char(20))
PK: (FSID)
CK: (FSName,FSAddress)
FK: CountryName references Country
Not Null: CountryName

University of British Columbia, Vancouver

Department of Computer Science

Country(CountryName: char(20), Population: real, Continent: char(20))

PK: (CountryName)

Not Null: Continent

GroceryStore(**FSID**: int, GSLicense: int, Departments: int)

PK: (FSID)

CK: (GSLicenseID)

FK: FSID references FoodServicesFrom

Restaurant(**FSID**: int, RLicense: int, Tables: int)

PK: (FSID)

CK: (RLicenseID)

FK: FSID references FoodServicesFrom

ContainIngredient(ProduceID: int, IName: char(20), Origin: char(20), MFGDate: date, Expiration: date, **FPName**: char(20))

PK: (ProduceID)

CK: (IName, Origin, MFG)

FK: FPName references FoodProductSell

Purchase(**ProduceID**: int, **FSID**: int, OrderID: int)

PK: (ProduceID)

CK: (OrderID)

FK: ProduceID references ContainIngredient

FSID references FoodServicesFrom

IncludeNutrition(**ProduceID**: int, ChemName: varchar, Benefit: varchar, Amount: int)

PK: (ProduceID, CName)

FK: ProduceID references ContainIngredient (on delete cascade)

AgriculturalDisease(ADName: char(20), Type: char(20), Solution: varchar, HealthEffect: varchar)

PK: (ADName)

Affect(**ADName**: char(20), **ProduceID**: int, Severeness: char(20), Date: date)

PK: (ADName, ProduceID)

FK: ADName references AgriculturalDisease

ProduceID references ContainIngredient

University of British Columbia, Vancouver

Department of Computer Science

HealthIssue(IssueName: char(20), Effect: varchar, Symptom: varchar)
PK: (IssueName)

Medicine(DIN: int, Manufacturer: char(20), DName: char(20), Expiration: date)
PK: (DIN)
CK: (Manufacturer, DName)

Cause(**FPName**: char(20), **IssueName**: char(20), EmergencyLevel: char(20))
PK: (FPName, IssueName)
FK: FPName references FoodProductSell
IssueName references HealthIssue

Treatment(**IssueName**: char(20), **DIN**: int, Dose: varchar)
PK: (IssueName, DIN)
FK: IssueName references HealthIssue
DIN references Medicine

FDs:

FoodProductSell(FPName, Description, Recipe, **FSID**)
FPName -> Description, Recipe, FSID

FoodServiceFrom(FSID, FSName, FSAddress, hours, **CountryName**)
FSID -> FSName, FSAddress, hours, CountryName
FSName, FSAddress -> FSID, hours, CountryName

Country(CountryName, Population, Continent)
CountryName -> Population, Continent

GroceryStore(**FSID**, GSLicense, Departments)
FSID -> GSLicense, Departments
GSLicense -> FSID, Departments

Restaurant(**FSID**, RLicense, Tables)
FSID -> RLicense, Tables
RLicense -> FSID, Tables

ContainIngredient(ProduceID, IName, Origin, MFGDate, Expiration, **FPName**)

University of British Columbia, Vancouver

Department of Computer Science

ProduceID -> IName, Origin, MFGDate, Expiration, FPName

IName, Origin, MFGDate -> ProduceID, Expiration, FPName

Origin, MFGDate -> Expiration

Purchase(**ProduceID**, **FSID**, OrderID)

ProduceID -> FSID, OrderID

OrderID -> ProduceID, FSID

IncludeNutrition(**ProduceID**, **ChemName**, Benefit, Amount)

ProduceID, ChemName -> Benefit, Amount

ChemName -> Benefit

AgriculturalDisease(**ADName**, Type, Solution, HealthEffect)

ADName -> Type, Treatment, HealthEffect

Type -> Solution

Affect(**ADName**, **ProduceID**, Severeness, Date)

ADName, ProduceID -> Severeness, Date

HealthIssue(**IssueName**, Symptom, Effect)

IssueName -> Symptom, Effect

Symptom -> Effect

Medicine(**DIN**, Manufacturer, DName, Expiration)

DIN -> Manufacturer, DName, Expiration

Manufacturer, DName -> DIN, Expiration

Cause(**FPName**, **IssueName**, EmergencyLevel)

FPName, IssueName -> EmergencyLevel

Treatment(**IssueName**, **DIN**, Dose)

IssueName, DIN -> Dose

Normalization

In Table ContainIngredient: Origin, MFG -> Expiration violate the BCNF
Origin, MFG+ = {Origin, MFG, Expiration}, so Origin, MFG is not superkey.

IngredientExpiration(**Origin: char(20), MFGDate: date**, Expiration: date)

PK: (Origin, MFGDate)

FK Origin, MFGDate reference ContainIngredient

ContainIngredient(**ProduceID: int**, IName: char(20), Origin: char(20), MFGDate: date,
FPName: char(20))

PK: (ProduceID)

CK: (IName, Origin, MFG)

FK: FPName references FoodProductSell

In Table IncludeNutrition: ChemName -> Benefit violate the BCNF
ChemName is not a superkey.

NutritionalBenefits(**ChemName** -> Benefit)

PK: (ChemName)

FK: ChemName reference IncludeNutrition

IncludeNutrition(**ProduceID: int**, **ChemName: varchar**, Amount: int)

PK: (ProduceID)

FK: ProduceID references ContainIngredient (on delete cascade)

In Table AgriculturalDiseaseL Type -> Solution violate the BCNF
Type is not superkey.

DiseaseSolutions(**Type: char(20)**, Solution: varchar)

PK: (Type)

FK: Type reference AgriculturalDisease

AgriculturalDisease(**ADName: char(20)**, Type: char(20), HealthEffect: varchar)

PK: (ADName)

University of British Columbia, Vancouver

Department of Computer Science

In Table HealthIssue: Symptom -> Effect violate BCNF
Symptom is not superkey.

SymptomEffects(Symptom varchar, Effect varchar)
PK(Symptom)
FK: Symptom reference HealthIssue

HealthIssue(IssueName: char(20), Symptom: varchar)
PK: (IssueName)

After Normalization we have tables:

FoodProductSell(FPName: char(20), Description: varchar, Recipe: varchar, **FSID**: int)
PK: (FPName)
FK: FSID references FoodServiceFrom

FoodServicesFrom(FSID: int, FSName: char(20), FSAddress: varchar, hours: varchar,
CountryName: char(20))
PK: (FSID)
CK: (FSName, FSAddress)
FK: CountryName references Country
Not Null: CountryName

Country(CountryName: char(20), Population: real, Continent: char(20))
PK: (CountryName)
Not Null: Continent

GroceryStore(**FSID**: int, GSLicense: int, Departments: int)
PK: (FSID)
CK: (GSLicenseID)
FK: FSID references FoodServicesFrom

Restaurant(**FSID**: int, RLicense: int, Tables: int)
PK: (FSID)
CK: (RLicenseID)
FK: FSID references FoodServicesFrom

University of British Columbia, Vancouver

Department of Computer Science

IngredientExpiration(**Origin: char(20)**, **MFGDate: date**, Expiration: date)

PK: (Origin, MFGDate)

FK Origin, MFGDate reference ContainIngredient

ContainIngredient(**ProduceID: int**, **IName: char(20)**, **Origin: char(20)**, **MFGDate: date**, **FPName: char(20)**)

PK: (ProduceID)

CK: (IName, Origin, MFG)

FK: FPName references FoodProductSell

Purchase(**ProduceID: int**, **FSID: int**, OrderID: int)

PK: (ProduceID)

CK: (OrderID)

FK: ProduceID references ContainIngredient

FSID references FoodServicesFrom

NutritionalBenefits(**ChemName** -> Benefit)

PK: (ChemName)

FK: ChemName reference IncludeNutrition

IncludeNutrition(**ProduceID: int**, **ChemName: varchar**, Amount: int)

PK: (ProduceID)

FK: ProduceID references ContainIngredient (on delete cascade)

DiseaseSolutions(**Type: char(20)**, Solution: varchar)

PK: (Type)

FK: Type reference AgriculturalDisease

AgriculturalDisease(**ADName: char(20)**, **Type: char(20)**, HealthEffect: varchar)

PK: (ADName)

Affect(**ADName**, **ProduceID**, Severeness, Date)

ADName, ProduceID -> Severeness, Date

SymptomEffects(**Symptom** varchar, Effect varchar)

PK(Symptom)

FK: Symptom reference HealthIssue

University of British Columbia, Vancouver

Department of Computer Science

HealthIssue(IssueName: char(20), Symptom: varchar)

PK: (IssueName)

Medicine(DIN: int, Manufacturer: char(20), DName: char(20), Expiration: date)

PK: (DIN)

CK: (Manufacturer, DName)

Cause(**FPName**: char(20), **IssueName**: char(20), EmergencyLevel: char(20))

PK: (FPName, IssueName)

FK: FPName references FoodProductSell

IssueName references HealthIssue

Treatment(**IssueName**: char(20), **DIN**: int, Dose: varchar)

PK: (IssueName, DIN)

FK: IssueName references HealthIssue

DIN references Medicine

SQL DDL Create Statements

```
CREATE TABLE Country (  
    CountryName CHAR(20) PRIMARY KEY,  
    Population REAL NOT NULL,  
    Continent CHAR(20) NOT NULL  
);
```

```
CREATE TABLE FoodServiceFrom (  
    FSID INT PRIMARY KEY,  
    FSName CHAR(20),  
    FSAddress VARCHAR,  
    hours VARCHAR,  
    CountryName CHAR(20) NOT NULL,  
    FOREIGN KEY (CountryName) REFERENCES Country(CountryName)  
);
```

University of British Columbia, Vancouver

Department of Computer Science

```
CREATE TABLE FoodProductSell (  
    FPName CHAR(20) PRIMARY KEY,  
    Description VARCHAR,  
    Recipe VARCHAR,  
    FSID INT,  
    FOREIGN KEY (FSID) REFERENCES FoodServiceFrom(FSID)  
);
```

```
CREATE TABLE GroceryStore (  
    FSID INT PRIMARY KEY,  
    GSLicense INT,  
    Departments INT,  
    FOREIGN KEY (FSID) REFERENCES FoodServiceFrom(FSID)  
);
```

```
CREATE TABLE Restaurant (  
    FSID INT PRIMARY KEY,  
    RLicense INT,  
    Tables INT,  
    FOREIGN KEY (FSID) REFERENCES FoodServiceFrom(FSID)  
);
```

```
CREATE TABLE ContainIngredient (  
    ProduceID INT PRIMARY KEY,  
    IName CHAR(20),  
    Origin CHAR(20),  
    MFGDate DATE,  
    FPName CHAR(20),  
    FOREIGN KEY (FPName) REFERENCES FoodProductSell(FPName)  
);
```

```
CREATE TABLE Purchase (  
    ProduceID INT,  
    FSID INT,  
    OrderID INT,  
    PRIMARY KEY (ProduceID),  
    FOREIGN KEY (ProduceID) REFERENCES ContainIngredient(ProduceID),  
    FOREIGN KEY (FSID) REFERENCES FoodServiceFrom(FSID)  
);
```

University of British Columbia, Vancouver

Department of Computer Science

```
CREATE TABLE IncludeNutrition (  
    ProduceID INT PRIMARY KEY,  
    ChemName VARCHAR,  
    Amount INT,  
    FOREIGN KEY (ProduceID) REFERENCES ContainIngredient(ProduceID) ON DELETE  
    CASCADE  
);
```

```
CREATE TABLE IngredientExpiration (  
    Origin CHAR(20),  
    MFGDate DATE,  
    Expiration DATE,  
    PRIMARY KEY (Origin, MFGDate),  
    FOREIGN KEY (Origin, MFGDate) REFERENCES ContainIngredient(Origin, MFGDate)  
);
```

```
CREATE TABLE DiseaseSolutions (  
    Type CHAR(20) PRIMARY KEY,  
    Solution VARCHAR  
);
```

```
CREATE TABLE AgriculturalDisease (  
    ADName CHAR(20) PRIMARY KEY,  
    Type CHAR(20),  
    HealthEffect VARCHAR,  
    FOREIGN KEY (Type) REFERENCES DiseaseSolutions(Type)  
);
```

```
CREATE TABLE Affect (  
    ADName CHAR(20),  
    ProduceID INT,  
    Severeness INT,  
    Date DATE,  
    PRIMARY KEY (ADName, ProduceID),  
    FOREIGN KEY (ADName) REFERENCES AgriculturalDisease(ADName),  
    FOREIGN KEY (ProduceID) REFERENCES ContainIngredient(ProduceID)  
);
```

University of British Columbia, Vancouver

Department of Computer Science

```
CREATE TABLE SymptomEffects (  
    Symptom VARCHAR PRIMARY KEY,  
    Effect VARCHAR  
);
```

```
CREATE TABLE HealthIssue (  
    IssueName CHAR(20) PRIMARY KEY,  
    Symptom VARCHAR,  
    FOREIGN KEY (Symptom) REFERENCES SymptomEffects(Symptom)  
);
```

```
CREATE TABLE Medicine (  
    DIN INT PRIMARY KEY,  
    Manufacturer CHAR(20),  
    DName CHAR(20),  
    Expiration DATE,  
    UNIQUE (Manufacturer, DName)  
);
```

```
CREATE TABLE Cause (  
    FPName CHAR(20),  
    IssueName CHAR(20),  
    EmergencyLevel CHAR(20),  
    PRIMARY KEY (FPName, IssueName),  
    FOREIGN KEY (FPName) REFERENCES FoodProductSell(FPName),  
    FOREIGN KEY (IssueName) REFERENCES HealthIssue(IssueName)  
);
```

```
CREATE TABLE Treatment (  
    IssueName CHAR(20),  
    DIN INT,  
    Dose VARCHAR,  
    PRIMARY KEY (IssueName, DIN),  
    FOREIGN KEY (IssueName) REFERENCES HealthIssue(IssueName),  
    FOREIGN KEY (DIN) REFERENCES Medicine(DIN)  
);
```

Insert Statements

// -- Inserting data into Country

```
INSERT INTO Country (CountryName, Population, Continent) VALUES ('CountryA',  
100000, 'Continent1');
```

```
INSERT INTO Country (CountryName, Population, Continent) VALUES ('CountryB',  
200000, 'Continent2');
```

```
INSERT INTO Country (CountryName, Population, Continent) VALUES ('CountryC',  
300000, 'Continent1');
```

```
INSERT INTO Country (CountryName, Population, Continent) VALUES ('CountryD',  
400000, 'Continent2');
```

```
INSERT INTO Country (CountryName, Population, Continent) VALUES ('CountryE',  
500000, 'Continent1');
```

// -- Inserting data into FoodServicesFrom

```
INSERT INTO FoodServicesFrom (FSID, FSName, FSAddress, hours, CountryName) VALUES  
(1, 'ServiceA', 'Address1', '9-5', 'CountryA');
```

```
INSERT INTO FoodServicesFrom (FSID, FSName, FSAddress, hours, CountryName) VALUES  
(2, 'ServiceB', 'Address2', '10-6', 'CountryB');
```

```
INSERT INTO FoodServicesFrom (FSID, FSName, FSAddress, hours, CountryName) VALUES  
(3, 'ServiceC', 'Address3', '11-7', 'CountryC');
```

```
INSERT INTO FoodServicesFrom (FSID, FSName, FSAddress, hours, CountryName) VALUES  
(4, 'ServiceD', 'Address4', '8-4', 'CountryD');
```

```
INSERT INTO FoodServicesFrom (FSID, FSName, FSAddress, hours, CountryName) VALUES  
(5, 'ServiceE', 'Address5', '12-8', 'CountryE');
```

```
INSERT INTO FoodServicesFrom (FSID, FSName, FSAddress, hours, CountryName) VALUES  
(6, 'ServiceF', 'Address6', '2-5', 'CountryA');
```

```
INSERT INTO FoodServicesFrom (FSID, FSName, FSAddress, hours, CountryName) VALUES  
(7, 'ServiceG', 'Address7', '1-6', 'CountryB');
```

```
INSERT INTO FoodServicesFrom (FSID, FSName, FSAddress, hours, CountryName) VALUES  
(8, 'ServiceH', 'Address8', '19-7', 'CountryC');
```

```
INSERT INTO FoodServicesFrom (FSID, FSName, FSAddress, hours, CountryName) VALUES  
(9, 'ServiceI', 'Address9', '8-3', 'CountryD');
```

```
INSERT INTO FoodServicesFrom (FSID, FSName, FSAddress, hours, CountryName) VALUES  
(10, 'ServiceJ', 'Address10', '20-8', 'CountryE');
```

University of British Columbia, Vancouver

Department of Computer Science

```
// -- Inserting data into GroceryStore
```

```
INSERT INTO GroceryStore (FSID, GSLicense, Departments) VALUES (1, 1001, 10);
```

```
INSERT INTO GroceryStore (FSID, GSLicense, Departments) VALUES (2, 1002, 12);
```

```
INSERT INTO GroceryStore (FSID, GSLicense, Departments) VALUES (3, 1003, 8);
```

```
INSERT INTO GroceryStore (FSID, GSLicense, Departments) VALUES (4, 1004, 15);
```

```
INSERT INTO GroceryStore (FSID, GSLicense, Departments) VALUES (5, 1005, 5);
```

```
// -- Inserting data into Restaurant
```

```
INSERT INTO Restaurant (FSID, RLicense, Tables) VALUES (6, 2001, 20);
```

```
INSERT INTO Restaurant (FSID, RLicense, Tables) VALUES (7, 2002, 25);
```

```
INSERT INTO Restaurant (FSID, RLicense, Tables) VALUES (8, 2003, 30);
```

```
INSERT INTO Restaurant (FSID, RLicense, Tables) VALUES (9, 2004, 10);
```

```
INSERT INTO Restaurant (FSID, RLicense, Tables) VALUES (10, 2005, 50);
```

```
// -- Inserting data into FoodProductSell
```

```
INSERT INTO FoodProductSell (FPName, Description, Recipe, FSID) VALUES ('ProductA',  
'Desc1', 'Recipe1', 1);
```

```
INSERT INTO FoodProductSell (FPName, Description, Recipe, FSID) VALUES ('ProductB',  
'Desc2', 'Recipe2', 2);
```

```
INSERT INTO FoodProductSell (FPName, Description, Recipe, FSID) VALUES ('ProductC',  
'Desc3', 'Recipe3', 3);
```

```
INSERT INTO FoodProductSell (FPName, Description, Recipe, FSID) VALUES ('ProductD',  
'Desc4', 'Recipe4', 4);
```

```
INSERT INTO FoodProductSell (FPName, Description, Recipe, FSID) VALUES ('ProductE',  
'Desc5', 'Recipe5', 5);
```

```
// -- Inserting data into IngredientExpiration
```

```
INSERT INTO IngredientExpiration (Origin, MFGDate, Expiration) VALUES ('Origin1',  
'2022-01-01', '2023-01-01');
```

```
INSERT INTO IngredientExpiration (Origin, MFGDate, Expiration) VALUES ('Origin2',  
'2022-02-01', '2023-02-01');
```

```
INSERT INTO IngredientExpiration (Origin, MFGDate, Expiration) VALUES ('Origin3',  
'2022-03-01', '2023-03-01');
```

University of British Columbia, Vancouver

Department of Computer Science

```
INSERT INTO IngredientExpiration (Origin, MFGDate, Expiration) VALUES ('Origin4',  
'2022-04-01', '2023-04-01');
```

```
INSERT INTO IngredientExpiration (Origin, MFGDate, Expiration) VALUES ('Origin5',  
'2022-05-01', '2023-05-01');
```

```
// -- Inserting data into ContainIngredient
```

```
INSERT INTO ContainIngredient (ProduceID, IName, Origin, MFGDate, FPName) VALUES  
(1, 'IngredientA', 'Origin1', '2022-01-01', 'ProductA');
```

```
INSERT INTO ContainIngredient (ProduceID, IName, Origin, MFGDate, FPName) VALUES  
(2, 'IngredientB', 'Origin2', '2022-02-01', 'ProductB');
```

```
INSERT INTO ContainIngredient (ProduceID, IName, Origin, MFGDate, FPName) VALUES  
(3, 'IngredientC', 'Origin3', '2022-03-01', 'ProductC');
```

```
INSERT INTO ContainIngredient (ProduceID, IName, Origin, MFGDate, FPName) VALUES  
(4, 'IngredientD', 'Origin4', '2022-04-01', 'ProductD');
```

```
INSERT INTO ContainIngredient (ProduceID, IName, Origin, MFGDate, FPName) VALUES  
(5, 'IngredientE', 'Origin5', '2022-05-01', 'ProductE');
```

```
// -- Inserting data into IncludeNutrition
```

```
INSERT INTO IncludeNutrition (ProduceID, ChemName, Amount) VALUES (1, 'ChemicalA',  
100);
```

```
INSERT INTO IncludeNutrition (ProduceID, ChemName, Amount) VALUES (2, 'ChemicalB',  
200);
```

```
INSERT INTO IncludeNutrition (ProduceID, ChemName, Amount) VALUES (3, 'ChemicalC',  
300);
```

```
INSERT INTO IncludeNutrition (ProduceID, ChemName, Amount) VALUES (4, 'ChemicalD',  
400);
```

```
INSERT INTO IncludeNutrition (ProduceID, ChemName, Amount) VALUES (5, 'ChemicalE',  
500);
```

```
// -- Inserting data into NutritionalBenefits
```

```
INSERT INTO NutritionalBenefits (ChemName, Benefit) VALUES ('ChemicalA', 'BenefitA');
```

```
INSERT INTO NutritionalBenefits (ChemName, Benefit) VALUES ('ChemicalB', 'BenefitB');
```

```
INSERT INTO NutritionalBenefits (ChemName, Benefit) VALUES ('ChemicalC', 'BenefitC');
```

```
INSERT INTO NutritionalBenefits (ChemName, Benefit) VALUES ('ChemicalD', 'BenefitD');
```


University of British Columbia, Vancouver

Department of Computer Science

```
INSERT INTO NutritionalBenefits (ChemName, Benefit) VALUES ('ChemicalE', 'BenefitE');
```

```
// -- Inserting data into DiseaseSolutions
```

```
INSERT INTO DiseaseSolutions (Type, Solution) VALUES ('TypeA', 'SolutionA');
```

```
INSERT INTO DiseaseSolutions (Type, Solution) VALUES ('TypeB', 'SolutionB');
```

```
INSERT INTO DiseaseSolutions (Type, Solution) VALUES ('TypeC', 'SolutionC');
```

```
INSERT INTO DiseaseSolutions (Type, Solution) VALUES ('TypeD', 'SolutionD');
```

```
INSERT INTO DiseaseSolutions (Type, Solution) VALUES ('TypeE', 'SolutionE');
```

```
// -- Inserting data into AgriculturalDisease
```

```
INSERT INTO AgriculturalDisease (ADName, Type, HealthEffect) VALUES ('DiseaseA',  
'TypeA', 'EffectA');
```

```
INSERT INTO AgriculturalDisease (ADName, Type, HealthEffect) VALUES ('DiseaseB',  
'TypeB', 'EffectB');
```

```
INSERT INTO AgriculturalDisease (ADName, Type, HealthEffect) VALUES ('DiseaseC',  
'TypeC', 'EffectC');
```

```
INSERT INTO AgriculturalDisease (ADName, Type, HealthEffect) VALUES ('DiseaseD',  
'TypeD', 'EffectD');
```

```
INSERT INTO AgriculturalDisease (ADName, Type, HealthEffect) VALUES ('DiseaseE',  
'TypeE', 'EffectE');
```

```
// -- Inserting data into Affect
```

```
INSERT INTO Affect (ADName, ProduceID, Severeness, Date) VALUES ('DiseaseA', 1,  
'High', '2024-06-01');
```

```
INSERT INTO Affect (ADName, ProduceID, Severeness, Date) VALUES ('DiseaseB', 2,  
'Medium', '2024-07-01');
```

```
INSERT INTO Affect (ADName, ProduceID, Severeness, Date) VALUES ('DiseaseC', 3, 'Low',  
'2024-08-01');
```

```
INSERT INTO Affect (ADName, ProduceID, Severeness, Date) VALUES ('DiseaseD', 4,  
'Critical', '2024-09-01');
```

```
INSERT INTO Affect (ADName, ProduceID, Severeness, Date) VALUES ('DiseaseE', 5,  
'Moderate', '2024-10-01');
```

University of British Columbia, Vancouver

Department of Computer Science

```
// -- Inserting data into SymptomEffects
```

```
INSERT INTO SymptomEffects (Symptom, Effect) VALUES ('SymptomA', 'EffectA');
```

```
INSERT INTO SymptomEffects (Symptom, Effect) VALUES ('SymptomB', 'EffectB');
```

```
INSERT INTO SymptomEffects (Symptom, Effect) VALUES ('SymptomC', 'EffectC');
```

```
INSERT INTO SymptomEffects (Symptom, Effect) VALUES ('SymptomD', 'EffectD');
```

```
INSERT INTO SymptomEffects (Symptom, Effect) VALUES ('SymptomE', 'EffectE');
```

```
// -- Inserting data into HealthIssue
```

```
INSERT INTO HealthIssue (IssueName, Symptom) VALUES ('IssueA', 'SymptomA');
```

```
INSERT INTO HealthIssue (IssueName, Symptom) VALUES ('IssueB', 'SymptomB');
```

```
INSERT INTO HealthIssue (IssueName, Symptom) VALUES ('IssueC', 'SymptomC');
```

```
INSERT INTO HealthIssue (IssueName, Symptom) VALUES ('IssueD', 'SymptomD');
```

```
INSERT INTO HealthIssue (IssueName, Symptom) VALUES ('IssueE', 'SymptomE');
```

```
// -- Inserting data into Medicine
```

```
INSERT INTO Medicine (DIN, Manufacturer, DName, Expiration) VALUES ('DIN001',  
'ManufacturerA', 'MedicineA', '2025-01-01');
```

```
INSERT INTO Medicine (DIN, Manufacturer, DName, Expiration) VALUES ('DIN002',  
'ManufacturerB', 'MedicineB', '2025-02-01');
```

```
INSERT INTO Medicine (DIN, Manufacturer, DName, Expiration) VALUES ('DIN003',  
'ManufacturerC', 'MedicineC', '2025-03-01');
```

```
INSERT INTO Medicine (DIN, Manufacturer, DName, Expiration) VALUES ('DIN004',  
'ManufacturerD', 'MedicineD', '2025-04-01');
```

```
INSERT INTO Medicine (DIN, Manufacturer, DName, Expiration) VALUES ('DIN005',  
'ManufacturerE', 'MedicineE', '2025-05-01');
```

```
// -- Inserting data into Cause
```

```
INSERT INTO Cause (FPName, IssueName, EmergencyLevel) VALUES ('ProductA', 'IssueA',  
'High');
```

```
INSERT INTO Cause (FPName, IssueName, EmergencyLevel) VALUES ('ProductB', 'IssueB',  
'Medium');
```

```
INSERT INTO Cause (FPName, IssueName, EmergencyLevel) VALUES ('ProductC', 'IssueC',  
'Low');
```

University of British Columbia, Vancouver

Department of Computer Science

```
INSERT INTO Cause (FPName, IssueName, EmergencyLevel) VALUES ('ProductD', 'IssueD', 'Critical');
```

```
INSERT INTO Cause (FPName, IssueName, EmergencyLevel) VALUES ('ProductE', 'IssueE', 'Moderate');
```

```
// -- Inserting data into Treatment
```

```
INSERT INTO Treatment (IssueName, DIN, Dose) VALUES ('IssueA', 'DIN001', 'DoseA');
```

```
INSERT INTO Treatment (IssueName, DIN, Dose) VALUES ('IssueB', 'DIN002', 'DoseB');
```

```
INSERT INTO Treatment (IssueName, DIN, Dose) VALUES ('IssueC', 'DIN003', 'DoseC');
```

```
INSERT INTO Treatment (IssueName, DIN, Dose) VALUES ('IssueD', 'DIN004', 'DoseD');
```

```
INSERT INTO Treatment (IssueName, DIN, Dose) VALUES ('IssueE', 'DIN005', 'DoseE');
```