**University of British Columbia, Vancouver**
Department of Computer Science

# CPSC 304 Project Cover Page

Milestone #: 1

Date: Feb 5th 2024

Group Number: 6

| Name | Student Number | CS Alias (Userid) | Preferred E-mail Address |
|---|---|---|---|
| Eun Ji Hwang | 63483804 | v5c7v@ugrad.cs.ubc.ca | eunji1120@outlook.com |
| Lavender Yu | 61960324 | q8k7c@ugrad.cs.ubc.ca | lavenderyu0113@gmail.com |
| An Zhou | 36008316 | i2l0z@ugrad.cs.ubc.ca | brico0532@gmail.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

**University of British Columbia, Vancouver**
Department of Computer Science

## ***Summary***

This project is meant to help the management of a hotpot restaurant by keeping track of its storage and sales. In this application, business managers can keep track of the sales per day per item in each branch and get updated information of the storage condition. Those information kept in this database helps managers to make efficient and reasonable decisions.

## ***ER diagram***

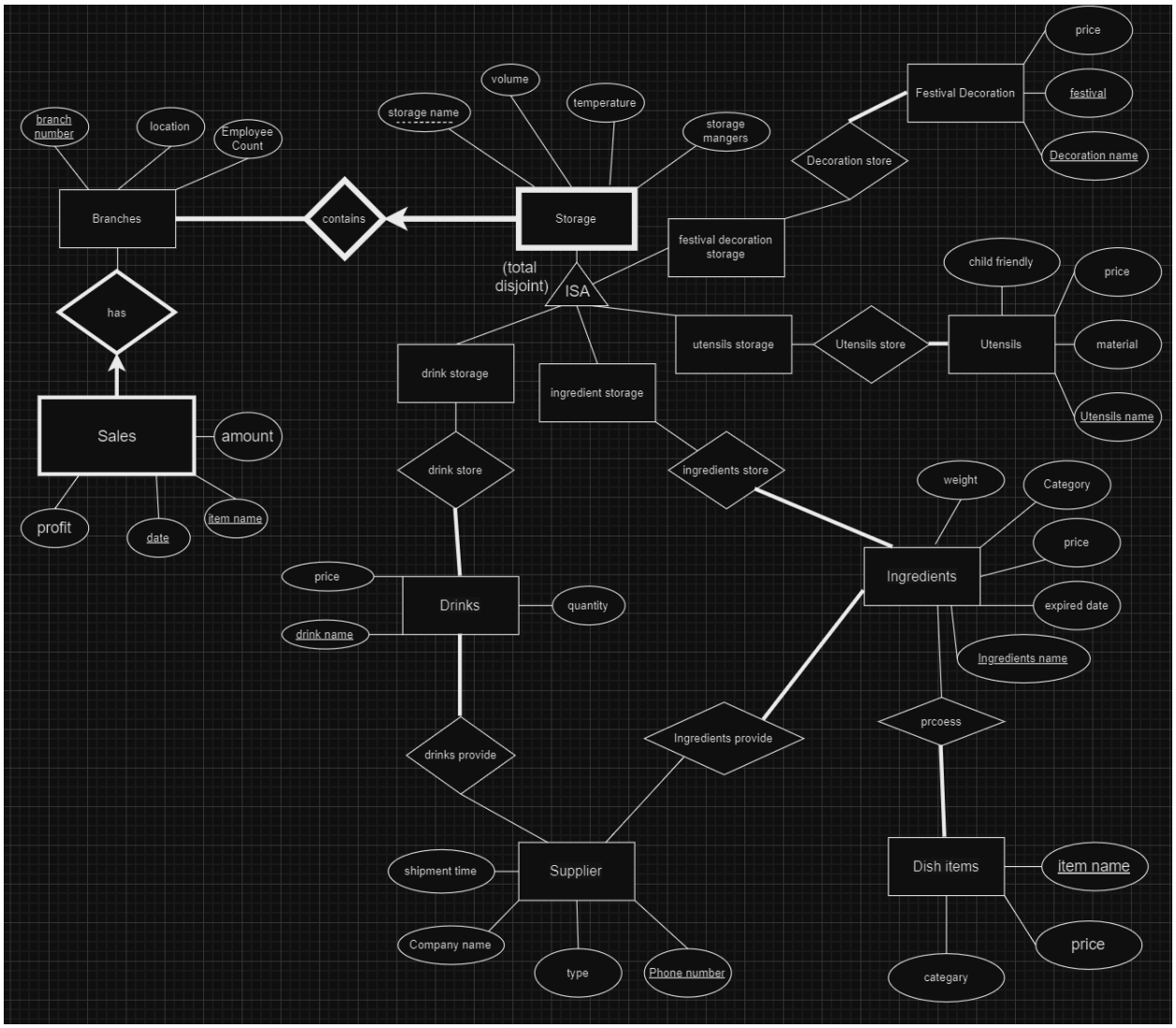
We changed the ER diagram based on the suggestions given by our project mentor. We have communicated with our mentor about most of the changes we made. Some of the major changes include:
1. Split storage into 3 entity sets using ISA to fix the mistake that existed in the original 'store' relation set. It is easier for implementation and avoids logical confusion.

2. Deleted unnecessary ISA relation and entity sets under 'dish items'. This is redundant information which can be displayed in an attribute.

3. Deleted messy relation set 'create' to rearranged entity set 'sales'. The relation causes confusion that is not able to be fixed.

4. Fixed some confusing attributes such as 'quantity', 'unit'. Those attributes might make sense or nor related

And some other minor changes to make the system more approachable to real-life restaurants.




## ***Schema***

*note: bold indicates foreign key, underline indicates primary key (as used in lectures)*

Branches (<u>branch number: int</u>, location: varchar(30), Employee Count: int) (location is not null and unique)
PK: branch number
CK: location
FK: N/A
NOT NULL: location
UNIQUE: location
*note: Employee Count should always be positive*

---

Drink Storage (**branch number: int**, storage name: varchar(30), volume:varchar(10), temperature: varchar(10) ,storage manager: varchar(30)) (temperature and volume are not null) (storage manager is a people who record the storage usage, we assume each storage manager has a unique name)
PK: branch number, storage name
CK: storage manager
FK:branch number
NOT NULL:temperature,volume
UNIQUE: storage manager
*Note : We are using varchar for volume and temperature since we also want to record the units that match with the number. (e.g., "500 ml", "2 L").*

Ingredient Storage (**branch number: int**, storage name:varchar(30), volume:varchar(10), temperature: varchar(10) , storage manager: varchar(30)) (temperature and volume are not null)
PK: branch number, storage name
CK: storage manager
FK:branch number
NOT NULL:temperature,volume
UNIQUE: storage manager
*Note : We are using varchar for volume and temperature since we also want to record the units that match with the number. (e.g., "500 ml", "2 L").*

Utensils Storage (**branch number: int**, storage name: varchar(30), volume:varchar(10), temperature: varchar(10) , storage manager:varchar(30))(temperature and volume are not null)
PK: branch number, storage name
CK: storage manager
FK:branch number
NOT NULL:temperature,volume
UNIQUE: storage manager
*Note : We are using varchar for volume and temperature since we also want to record the units that match with the number. (e.g., "500 ml", "2 L").*

Festival Decoration storage (**branch number:int**, storage name: varchar(30),volume:varchar(10), temperature: varchar(10) ,storage manager:varchar(30) )(temperature and volume are not null)
PK: branch number, storage name
CK: storage manager
FK:branch number
NOT NULL:temperature,volume
UNIQUE: storage manager

---

*Note : We are using varchar for volume and temperature since we also want to record the units that match with the number. (e.g., "500 ml", "2 L").*

Drink store(**storage name:varchar(30)**, **drink name:varchar(30), branch number:int)**
PK: storage name,drink name, branch number
CK: NA
FK: storage name,drink name, branch number
NOT NULL:N/A
UNIQUE: N/A

Ingredient store(**storage name:varchar(30), Ingredients name:varchar(30),branch number:varchar(30))**
PK: storage name,Ingredient name, branch number
CK: N/A
FK:storage name,Ingredient name, branch number
NOT NULL:N/A
UNIQUE: N/A

Utensils store(**storage name:varchar(30), Utensils name:varchar(30),branch number:int)**
PK: storage name,Utensils name, branch number
CK: N/A
FK:storage name,Utensils name, branch number
NOT NULL:N/A
UNIQUE: N/A

Festival Decoration store(**storage name:varchar(30), Festival Decoration name:varchar(30),branch number:int)**
PK: storage name,Festival Decoration name, branch number
CK: NA
FK:storage name,Festival Decoration name, branch number
NOT NULL:N/A
UNIQUE: N/A

Drinks(drink name:varchar(30),price:int,quantity:int) (price and quantity are not null)
PK: drink name
CK: N/A
FK:N/A
NOT NULL:price,quantity
UNIQUE: N/A

---

Ingredients(ingredients name:varchar(30), expired date: date, price:int, quantity:int, category:varchar(30))(expired date,price,quantity,category are not null)
PK: ingredients name
CK: N/A
FK:N/A
NOT NULL:price,quantity,expired date,Category
UNIQUE: N/A

Utensils(Utensils name:varchar(30), material:varchar(30),price:int, child_friendly:boolean)(material,price are not null)
PK: Utensils name
CK: NA
FK:NA
NOT NULL:material,price
UNIQUE: NA

Festival Decoration(Decoration name:varchar(30),festiva:varchar(30),price:int)(price is not null)
PK: Decoration name
CK: NA
FK:NA
NOT NULL:price
UNIQUE: NA

Supplier(Phone number:int,shipment time:int, Company name:varchar(30),type:varchar(30))(shipment time is not null,Company name is not null and unique)
PK: phone number
CK: Company name
FK:NA
NOT NULL:shipment time,Company name
UNIQUE: Company name

Drinks provide(**drink name:varchar(30), Phone number:int**)
PK: drink name,Phone number
CK: NA
FK:drink name,Phone number
NOT NULL:NA
UNIQUE: NA

Ingredients provide(**ingredients name:varchar(30), Phone number:int**)
PK:ingredients name,Phone number
CK: NA
FK:ingredients name,Phone number
NOT NULL:NA
UNIQUE: NA

Process(**Ingredients name:varchar(30), item name:CHAR(30)**)
PK: ingredients name,item name
CK: NA
FK: ingredients name,item name
NOT NULL:NA
UNIQUE: NA

Dish_items(item name:varchar(30),category:varchar(30), price:int) (category and price are not null)
PK: item name
CK: NA
FK: NA
NOT NULL:category,price
UNIQUE: NA

Has_Sales((**branch number:int,** item name :varchar(30),date:date,amount:int,profit:int)
PK: branch number, item name, date
CK: NA
FK: branch number
NOT NULL:NA
UNIQUE: NA

## ***Functional Dependencies (FDs)***

Branches:

branch number → location, Employee Count

location (candidate Key)-> Employee Count, branch number

Drink Storage:
branch number, storage name → volume, temperature, storage manager

storage name → volume, temperature

Ingredients Storage:
branch number, storage name → volume, temperature, storage manager
storage name → volume, temperature

Utensils Storage:
branch number, storage name → volume, temperature, storage manager
storage name → volume, temperature

Festival Decoration Storage:
branch number, storage name → volume, temperature, storage manager
storage name → volume, temperature

*(Note: In our practical case, every branch is using the same set of fridge/cabinet, thus the storage name determines volume and temperature.)*

Drinks
drink name → price, quantity

Ingredients
ingredients name → expired date, price, quantity, Category

Utensils
utensils name → material, price, child friendly
material → child friendly

Festival Decoration
festival, Decoration name → price

Supplier:
phone number→ company name, type, shipment time
company name (candidate key) → shipment time, type, phone number

Dish items:
item name → category, price

Has_Sales:
branch_number, item name, date → amount, profit

(different branch might have different sales price so might have different profit)

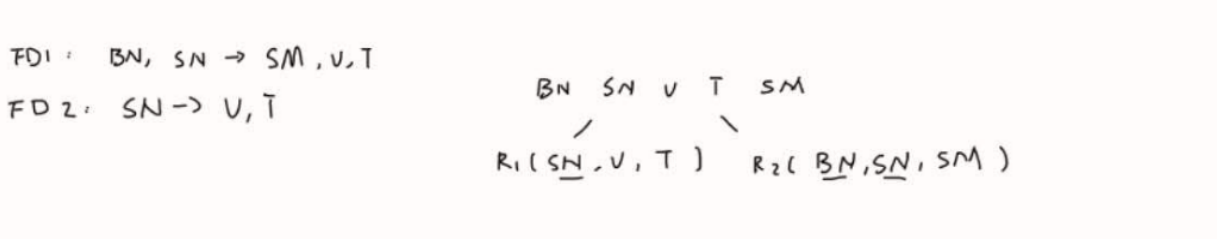
Note: we are not writing FDs for relation sets here because there are no attributes in our relation sets, thus all of the FDs will be trivial.

## ***Normalization***

We decompose each table into BCNF.

Branches（branch number: int, location: CHAR(30), Employee Count: int）
branch number → location, Employee Count
location (candidate Key)-> Employee Count, branch number
Keys: branch number, location
FD1 and FD2 does not violet BCNF, same from the previous step
Branch(branch number: int, location: CHAR(30), Employee Count: int)
primary key: branch number
candidate key: location
NOT NULL: location
UNIQUE: location
Employee Count should always be positive

Drink Storage（**branch number: int**, storage name: CHAR(30), volume:CHAR(10), temperature: CHAR(10) ,storage manager: CHAR(30) )
branch number, storage name → volume, temperature, storage manager
storage_name → volume, temperature
Keys: (branch number, storage name)

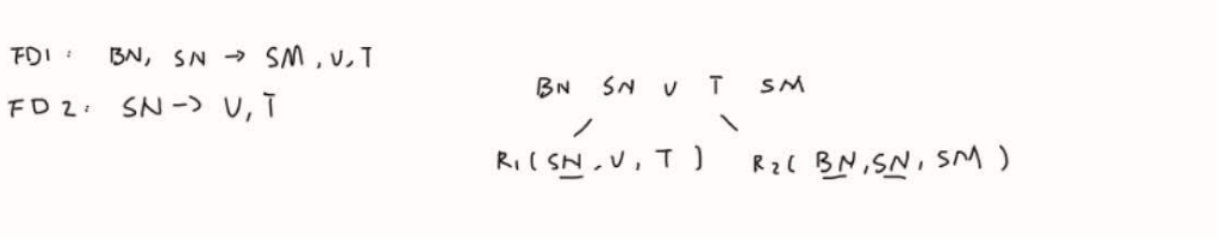



*notation: branch number(BN), storage name(SN),volume(V), temperature(T), storage manager(SM)*
Decomposition steps: FD2 violates BCNF
Then, we split it into R1 and R2.
Now both tables do not violate the BCNF.

R1=Drink Storage Condition Table (storage name: varchar(30), volume:varchar(10), temperature: varchar(10))
primary key:storage_name

R2=Drink Storage Belonging Table(**branch number: int**, (**branch number: int**, **storage_name**: varchar(30), storage manager: varchar(30))
primary key:storage_name, branch_number
foreign key: branch number,storage_name
unique: storage_name

Ingredients Storage（**branch number: int**, storage name: CHAR(30), volume:CHAR(10), temperature: CHAR(10) ,storage manager: CHAR(30) )
branch number, storage name → volume, temperature, storage manager
storage_name → volume, temperature
Keys: (branch number, storage name)




*notation: branch number(BN), storage name(SN),volume(V), temperature(T), storage manager(SM)*
Decomposition steps: FD2 violates BCNF
Then, we split it into R1 and R2.
Now both tables do not violate the BCNF.

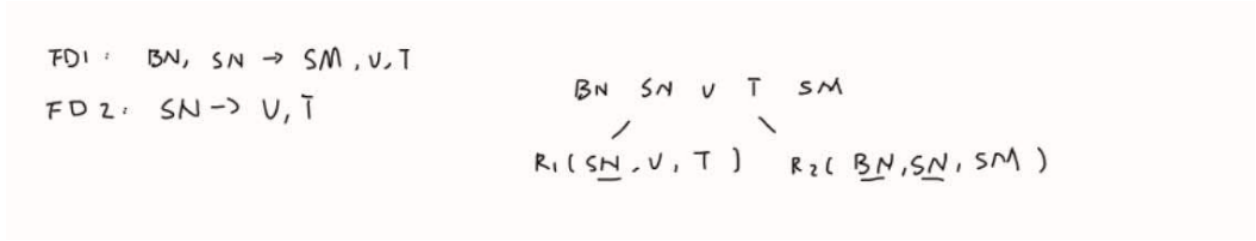
R1=Ingredients Storage Condition Table (storage name: varchar(30), volume:varchar(10), temperature: varchar(10))
primary key:storage_name

R2=Ingredients Storage Belonging Table (**branch number: int**, **storage_name**: varchar(30), storage manager: varchar(30))
primary key:storage_name, branch_number
foreign key: branch number,storage_name
unique: storage_name

**University of British Columbia, Vancouver**
Department of Computer Science

---

Utensils Storage（**branch number: int**, storage name: CHAR(30), volume:CHAR(10), temperature: CHAR(10) ,storage manager: CHAR(30) )
branch number, storage name → volume, temperature, storage manager
storage_name → volume, temperature
Keys: (branch number, storage name)

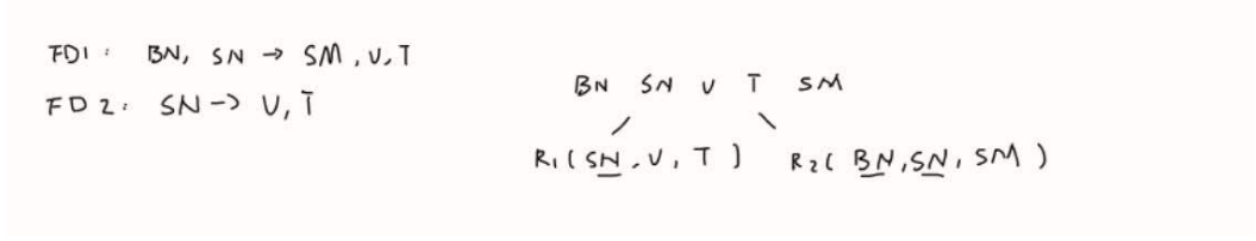



*notation: branch number(BN), storage name(SN),volume(V), temperature(T), storage manager(SM)*
Decomposition steps: FD2 violates BCNF
Then, we split it into R1 and R2.
Now both tables do not violate the BCNF.

R1=Utensils Storage Condition Table (storage name: varchar(30), volume:varchar(10), temperature: varchar(10))
primary key:storage_name

R2=Utensils Storage Belonging Table(**branch number: int**, **storage_name**: varchar(30), storage manager: varchar(30))
primary key:storage_name, branch_number
foreign key: branch number,storage_name
unique: storage_name

Festival Decoration Storage:
Festival Decoration Storage (**branch number: int**, storage name: CHAR(30), volume:CHAR(10),
temperature: CHAR(10) ,storage manager: CHAR(30) )
branch number, storage name → volume, temperature, storage manager
storage_name → volume, temperature



---

*notation: branch number(BN), storage name(SN),volume(V), temperature(T), storage manager(SM)*
Decomposition steps: FD2 violates BCNF
Then, we split it into R1 and R2.
Now both tables do not violate the BCNF.

R1=Festival Decoration Storage Condition Table (storage name: varchar(30), volume:varchar(10), temperature: varchar(10))
primary key:storage_name

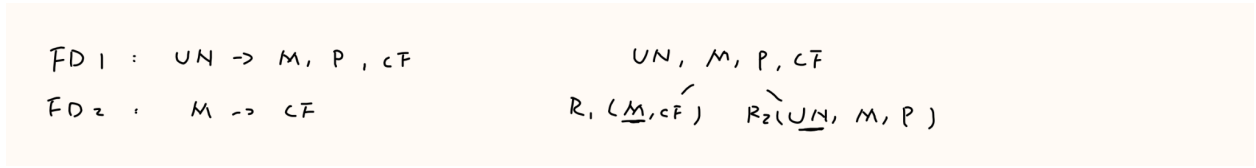
R2=Festival Decoration Storage Belonging Table(**branch number: int**, **storage_name**: varchar(30), storage manager: varchar(30))
primary key:storage_name, branch_number
foreign key: branch number,storage_name
unique: storage_name

Drinks
drink name → price, quantity
FD1 does not violet BCNF, same from the previous step

Drinks(drink name:varchar(30),price:int,quantity:int) (price and quantity are not null)
PK: drink name
CK: NA
FK:NA
NOT NULL:price,quantity
UNIQUE: NA

Ingredients
ingredients name → expired date, price, quantity, Category
FD1 does not violet BCNF, same from the previous step
Ingredients(ingredients name:varchar(30), expired date: date, price:int, quantity:int, category:varchar(30))(expired date,price,quantity,category are not null)
PK: ingredients name
CK: NA
FK:NA
NOT NULL:price,quantity,expired date,Category
UNIQUE: NA

Utensils
utensils name → material, price, child friendly
material -> child friendly




utensils name(UN), material(M),price(P),child friendly(CF)
Decomposition steps: FD2 violates BCNF, Then, we split it into Protection for Children Table and Utensils Information Table.
We found that the both Prevention of Allergies Table and Ingredients Information Table do satisfy the BCNF
After that, all two tables do not violate the BCNF.

R1= Protection for Children Table(material:varchar(30), child_friendly:boolean)
primary key:Material
candidate key:
foreign key:

R2=Utensils Information Table(Utensils name:varchar(30), **material:varchar(30)**,price:int, child_friendly:boolean)
primary key:Utensils name
candidate key:
foreign key: material
NOT NULL:material, price
UNIQUE: NA

Festival Decoration
festival, Decoration name → price
FD1 does not violet BCNF, same from the previous step

Festival Decoration(Decoration name:varchar(30),festiva:varchar(30),price:int)(price is not null)
PK: Decoration name
CK: NA
FK:NA
NOT NULL:price
UNIQUE: NA

Supplier:

phone number-> company name, type, shipment time
company name (candidate key) → shipment time, type, phone number
FD1 and FD2 does not violet BCNF, same from the previous step

Supplier(Phone number:int,shipment time:int, Company name:varchar(30),type:varchar(30))(shipment time is not null,Company name is not null and unique)
PK: phone number
CK: Company name
FK:NA
NOT NULL:shipment time,Company name
UNIQUE: Company name

Dish items:
item name → category, price
FD1 does not violet BCNF, same from the previous step
Dish_items(item name:varchar(30),category:varchar(30), price:int) (category and price are not null)
PK: item name
CK: NA
FK: NA
NOT NULL:category,price
UNIQUE: NA

Has_Sales:
item name, branch number, date → amount, profit
FD1 does not violet BCNF, same from the previous step
Has_Sales((**branch number:int,** item name :varchar(30),date:date,amount:int,profit:int)
PK: branch number, item name, date
CK: NA
FK: branch number
NOT NULL:NA
UNIQUE: NA

All relationships that have trivial relations do not violate BCNF, as is the case with the previous step.
e.g.: for drink_provide
drink name, phone number-> drink name , phone number

Drink store(**storage name:varchar(30)**, **drink name:varchar(30), branch number:int)**
PK: storage name,drink name, branch number
CK: NA
FK:storage name,drink name, branch number
NOT NULL:NA
UNIQUE: NA

Ingredient store(**storage name:varchar(30), Ingredients name:varchar(30),branch number:varchar(30))**
PK: storage name,Ingredient name, branch number
CK: NA
FK:storage name,Ingredient name, branch number
NOT NULL:NA
UNIQUE: NA

Utensils store(**storage name:varchar(30), Utensils name:varchar(30),branch number:int)**
PK: storage name,Utensils name, branch number
CK: NA
FK:storage name,Utensils name, branch number
NOT NULL:NA
UNIQUE: NA

Festival Decoration store(**storage name:varchar(30), Festival Decoration name:varchar(30),branch number:int)**
PK: storage name,Festival Decoration name, branch number
CK: NA
FK:storage name,Festival Decoration name, branch number
NOT NULL:NA
UNIQUE: NA

Drinks provide(**drink name:varchar(30), Phone number:int**)
PK: drink name,Phone number
CK: NA
FK:drink name,Phone number
NOT NULL:NA
UNIQUE: NA

Ingredients provide(**ingredients name:varchar(30), Phone number:int**)
PK:ingredients name,Phone number
CK: NA
FK:ingredients name,Phone number
NOT NULL:NA
UNIQUE: NA

Process(**Ingredients name:varchar(30), item name:CHAR(30)**)
PK: ingredients name,item name
CK: NA
FK: ingredients name,item name
NOT NULL:NA
UNIQUE: NA

## ***SQL DDL statements***

```
CREATE TABLE Branches (
  branch_number int PRIMARY KEY,
  location VARCHAR(30) NOT NULL,
  Employee_Count INT CHECK, (Employee_Count >= 0)
  UNIQUE(location)
);

CREATE TABLE Drink_Storage_Condition (
  storage_name VARCHAR(30) PRIMARY KEY,
  volume VARCHAR(10) NOT NULL,
  temperature VARCHAR(10) NOT NULL
);

CREATE TABLE Drink_Storage_Belonging (
  branch_number INT,
  storage_name VARCHAR(30) UNIQUE,
  storage_manager VARCHAR(30),
 PRIMARY KEY(branch_number, storage_manager)
  FOREIGN KEY (branch_number)
```

```
REFERENCES Branches(branch_number)
    ON DELETE CASCADE
    ON UPDATE CASCADE
);

CREATE TABLE Ingredients_Storage_Condition (
  storage_name VARCHAR(30) PRIMARY KEY,
  volume VARCHAR(10) NOT NULL,
  temperature VARCHAR(10) NOT NULL
);

CREATE TABLE Ingredients_Storage_Belonging (
  branch_number INT,
  storage_name VARCHAR(30) UNIQUE,
  storage_manager VARCHAR(30),
 PRIMARY KEY(branch_number, storage_manager)
  FOREIGN KEY (branch_number)
REFERENCES Branches(branch_number)
    ON DELETE CASCADE
    ON UPDATE CASCADE
);


CREATE TABLE Untensils_Storage_Condition (
  storage_name VARCHAR(30) PRIMARY KEY,
  volume VARCHAR(10) NOT NULL,
  temperature VARCHAR(10) NOT NULL
);

CREATE TABLE Untensils_Storage_Belonging (
  branch_number INT,
  storage_name VARCHAR(30) UNIQUE,
  storage_manager VARCHAR(30),
 PRIMARY KEY(branch_number, storage_manager)
  FOREIGN KEY (branch_number)
REFERENCES Branches(branch_number)
    ON DELETE CASCADE
    ON UPDATE CASCADE
);
```

```
CREATE TABLE Festival_Decoration_Storage_Condition (
  storage_name VARCHAR(30) PRIMARY KEY,
  volume VARCHAR(10) NOT NULL,
  temperature VARCHAR(10) NOT NULL
);

CREATE TABLE Festival_Decoration_Storage_Belonging (
  branch_number INT,
  storage_name VARCHAR(30) UNIQUE,
  storage_manager VARCHAR(30),
 PRIMARY KEY(branch_number, storage_manager)
  FOREIGN KEY (branch_number)
REFERENCES Branches(branch_number)
    ON DELETE CASCADE
    ON UPDATE CASCADE
);

CREATE TABLE Drink_store (
  storage_name VARCHAR(30),
  branch_number INT,
  drink_name VARCHAR(30),
  PRIMARY KEY(branch_number, storage_name, drink_name),
  FOREIGN KEY (branch_number, storage_name) REFERENCES Drink_Storage_Belonging
(branch_number, storage_name)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  FOREIGN KEY (drink_name) REFERENCES Drinks(drink_name)
    ON DELETE CASCADE
    ON UPDATE CASCADE
);

CREATE TABLE Ingredient_store (
  storage_name VARCHAR(30),
  branch_number INT,
  ingredients_name VARCHAR(30)
  PRIMARY KEY(branch_number, storage_name, ingredients_name),
  FOREIGN KEY (branch_number, storage_name)
          REFERENCES Ingredients_Storage_Belonging (branch_number, storage_name)
```

```
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  FOREIGN KEY (drink_name) REFERENCES Ingredients(ingredients_name)
    ON DELETE CASCADE
    ON UPDATE CASCADE
);

CREATE TABLE Utensils_store (
  storage_name VARCHAR(30),
  branch_number INT,
  Utensils_name VARCHAR(30),
  PRIMARY KEY(branch_number, storage_name, Utensils_name),
  FOREIGN KEY (branch_number, storage_name) REFERENCES Utensils_Storage_Belonging
(branch_number, storage_name)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  FOREIGN KEY (Utensils_name) REFERENCES Utensils(Utensils_name)
    ON DELETE CASCADE
    ON UPDATE CASCADE
);

CREATE TABLE Fesitval_Decoration_store (
  storage_name VARCHAR(30),
  branch_number INT,
 Fesitval_Decoration_name VARCHAR(30),
  PRIMARY KEY(branch_number, storage_name, Fesitval_Decoration_name),
  FOREIGN KEY (branch_number, storage_name) REFERENCES
Fesitval_Decoration_Storage_Belonging (branch_number, storage_name)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  FOREIGN KEY (Fesitval_Decoration_name) REFERENCES
Fesitval_Decoration(Fesitval_Decoration_name)
    ON DELETE CASCADE
    ON UPDATE CASCADE
);

CREATE TABLE Drinks (
  drink_name VARCHAR(30),
  price INT NOT NULL,
  quantity INT NOT NULL,
```

```
  PRIMARY KEY (drink_name)
);

CREATE TABLE Ingredients (
  ingredients_name VARCHAR(30),
  expired_date DATE NOT NULL,
  price INT NOT NULL,
  quantity INT NOT NULL,
  category VARCHAR(30) NOT NULL,
  PRIMARY KEY (ingredients_name)
);

CREATE TABLE Protection_for_Children (
  material VARCHAR(30) PRIMARY KEY,
  child_friendly BOOLEAN
);

CREATE TABLE Utensils_Information (
  utensils_name VARCHAR(30)PRIMARY KEY,
  material VARCHAR(30) NOT NULL,
  price INT NOT NULL,
  FOREIGN KEY (material) REFERENCES Protection_for_Children(material)
    ON DELETE RESTRICT
    ON UPDATE CASCADE
);

CREATE TABLE Festival_Decoration (
  decoration_name VARCHAR(30),
  festival VARCHAR(30),
  price INT NOT NULL,
  PRIMARY KEY (decoration_name)
);

CREATE TABLE Supplier (
  phone_number INT,
  shipment_time INT NOT NULL,
  company_name VARCHAR(30) NOT NULL,
  type VARCHAR(30),
  PRIMARY KEY (phone_number),
  UNIQUE(company_name)
```

```
);


CREATE TABLE Drinks_Provide (
  drink_name VARCHAR(30),
  Phone_number INT,
  PRIMARY KEY(drink_name,Phone_number),
  FOREIGN KEY (drink_name) REFERENCES Drinks(drink_name),
    ON DELETE CASCADE
    ON UPDATE CASCADE
  FOREIGN KEY (drink_name) REFERENCES Supplier( Phone_number)
    ON DELETE CASCADE
    ON UPDATE CASCADE
);


CREATE TABLE Ingredients_Provide (
  ingredients_name VARCHAR(30),
  Phone_number INT,
  PRIMARY KEY(ingredients_name,name,type),
  FOREIGN KEY (ingredients_name) REFERENCES Ingredients(ingredients_name),
    ON DELETE CASCADE
    ON UPDATE CASCADE
  FOREIGN KEY (drink_name) REFERENCES Supplier( Phone_number)
    ON DELETE CASCADE
    ON UPDATE CASCADE
);

CREATE TABLE Process (
  ingredients_name VARCHAR(30),
  item_name VARCHAR(30),
  PRIMARY KEY(ingredients_name,item_name),
  FOREIGN KEY (ingredients_name) REFERENCES Ingredients(ingredients_name)
    ON DELETE CASCADE
    ON UPDATE CASCADE
  FOREIGN KEY (item_name) REFERENCES Dish_Items(item_name)
    ON DELETE CASCADE
    ON UPDATE CASCADE
);
```

```
CREATE TABLE Dish_Items (
  item_name VARCHAR(30),
  category VARCHAR(30) NOT NULL,
  price INT NOT NULL,
  PRIMARY KEY (item_name)
);

CREATE TABLE Has_Sales (
  branch_number INT,
  item_name VARCHAR(30),
  date DATE,
  amount INT,
  profit INT,
  PRIMARY KEY(branch_number,item_name,date),
  FOREIGN KEY (branch_number) REFERENCES Branches
    ON DELETE CASCADE
    ON UPDATE CASCADE
);
```

## ***INSERT (populate data)***

```
INSERT INTO Branches(branch number, location, Employee_Count)
VALUES
   (8204, '1194 Saint-Catherine St W, Montreal, Quebec', 45),
   (3252, '3204 W Broadway, Vancouver, BC', 43),
   (3243, '5890 No. 3 Rd Room 200, Richmond, BC', 46),
   (1332, '5328 Hwy 7, Markham, ON', 40),
   (1294, '237 Yonge St, Toronto, ON', 39),
   (1124, '1571 Sandhurst Cir #106F, Scarborough, ON', 48);

INSERT INTO Drink_Storage_Condition (storage_name, volume, temperature)
VALUES
('large drink fridge', '30L', '3℃'),
('medium drink fridge', '25L'. '3℃'),
('small drink fridge', '15L', '3℃'),
('large drink cabinet', '20L', 'room temperature'),
('small drink cabinet', '10L', 'room temperature');

INSERT INTO Drink_Storage_Belonging (branch_number, storage_name, storage_manager)
VALUES
```

(8204, 'large drink fridge', 'Kevin.L'),
(8204, 'medium drink fridge', 'Leo.Z'),
(3243, 'medium drink fridge','Dora.Z',),
(1332, 'small drink cabinet', 'Nott.F'),
(1294, 'small drink cabinet', 'Oscar.M')
(1124, 'small drink cabinet', 'Collin.T');

INSERT INTO Ingredients_Storage_Condition (storage_name, volume, temperature)
VALUES
('pre-made food fridge', '45L', '-12℃'),
('spice rack', '3L', 'room temperature'),
('vegetable fridge', '40L', '3℃'),
('sauce cupboard', '5L', '3℃'),
('protein freezer', '80L', '-12℃'),
('seafood cooler', '50L', '-12℃');

INSERT INTO Ingredients_Storage_Belonging(branch_number, storage_name,
storage_manager)
VALUES
(8204, 'pre-made food fridge', 'Mark.D'),
(8204, 'spice rack', 'Erica.P'),
(3243, 'vegetable fridge', 'Angle.S'),
(1332, 'vegetable fridge', 'John.S'),
(1294, 'vegetable fridge', 'Rachel.Y'),
(1124, 'vegetable fridge', 'Daniel.A');

INSERT INTO Utensils_Storage_Condition (storage_name, volume, temperature)
VALUES
('small plate cabinet', '10L', 'room temperature'),
('large plate cabinet', '20L', 'room temperature'),
('tableware cabinet', '5L', 'room temperature'),
('pot and wok cabinet', '10L', 'room temperature'),
('cleaning tools storage', '50L', 'room temperature');

INSERT INTO Untensils_Storage_Belonging (storage_manager, branch_number)
VALUES
('Lucas.L', 8204),
('Michael.L', 3252),
('Jerry.F', 3243),
('Sophia.S', 1332),

('Amber.D', 1294);

INSERT INTO Festival_Decoration_Storage_Condition (storage_name, volume, temperature)
VALUES
('mid-autumn decoration', '5L', 'room temperature'),
('new year decoration', '5L', 'room temperature'),
('christmas decoration', '5L', "room temperature'),
('lunar new year decoration', '5L', 'room temperature'),
('valentine's day decoration', '5L','room temperature');

INSERT INTO Festival_Decoration_Storage_Belonging (branch_number, storage_name, storage_manager)
VALUES
(8204, 'mid-autumn decoration', 'Tim.H'),
(8204, 'christmas decoration', 'Tim.H'),
(3243, 'valentine's day decoration','Morton.W'),
(1332, 'valentine's day decoration', 'Steven.S',),
(1294, 'valentine's day decoration', 'Bill.N',)
(1124, 'valentine's day decoration', 'Nick.R' );

INSERT INTO Drink_store(storage_name, branch_number, drink_name)
VALUES
('large drink fridge', 3243, 'Bubble Tea'),
('medium drink fridge', 3243, 'Iced Lemon Tea'),
('small drink fridge', 3243, 'Mango Smoothie'),
('large drink cabinet', 3243, 'Green Tea'),
('small drink cabinet", 3252, 'Watermelon Juice');

INSERT INTO Ingredient_store(storage_name, branch_number, ingredients_name)
VALUES
('spice rack', 3243, 'Chili Powder'),
('vegetable fridge', 3243, 'Carrots'),
('sauce cupboard', 3252, 'Soy Sauce'),
('protein freezer', 3252, 'Beef Slices'),
('seafood cooler', 3252, 'Shrimp');

INSERT INTO Utensils_store(storage_name, branch_number, Utensils_name)
VALUES
('plate cabinet 1', 3252, 'large white plates'),
('plate cabinet 2', 3252, 'small dishes'),

```
('tableware cabinet', 3243, 'forks'),
('pot and wok cabinet', 3243, 'hotpot'),
('cleaning storage', 3252, 'mop');

INSERT INTO Festival_Decoration_store(storage_name, branch_number,
Fesitval_Decoration_name)
VALUES
   ('mid-autumn decoration', 3252, 'Colorful Lanterns'),
   ('christmas decoration', 3252, ‘Tinsel Garland’'),
   ('christmas decoration', 3243, ‘Paper Snowflakes’'),
   ('valentine's day decoration', 3243, 'Party Streamers'),
   ('lunar new year decoration', 3243, 'Festive Banners');

INSERT INTO Drinks(drink_name, price, quantity)
VALUES
   ('bubble tea', 5, 100),
   ('iced lemon tea', 4, 80),
   ('mango smoothie', 6, 120),
   ('green tea latte', 5, 90),
   ('watermelon juice', 4, 110);


INSERT INTO Ingredients_Information (ingredients_name, expire_date, price, quantity,
category)
VALUES
   ('chili powder', 2025-10-10, 8, 10, 'vegetarian/halal'),
   ('carrots', 2024-3-2, 3, 4, 'vegetarian/halal'),
   ('soy sauce', 2024-5-1, 7, 13, 'vegetarian/halal'),
   ('beef slices', 2024-2-29, 12, 40, 'halal'),
   ('shrimp', 2024-3-15, 12, 15, 'vegetarian/halal');

INSERT INTO Protection_for_Children  (material, child_friendly)
VALUES
('stainless steel', TRUE),
('plastic', FALSE),
('bamboo', TRUE),
('glass', FALSE),
('ceramic', FALSE);

INSERT INTO Utensils_Information (utensils_name, material, price)
```

```
VALUES
('large white plates', 'ceramics', 7 ),
('small dishes', 'ceramics', 3),
('forks', 'stainless steel', 1),
('hotpot', 'stainless steel', 12),
('mop', 'plastic and fibre', 5);

INSERT INTO Festival_Decoration(decoration_name, festival, price)
VALUES
   ('Colorful Lanterns', 'mid-autumn', 20),
   ('Tinsel Garland', 'Christmas', 15),
   ('Paper Snowflakes', 'Christmas', 10),
   ('Party Streamers', 'valentine's day', 12),
   ('Festive Banners', 'lunar new year'', 18);

INSERT INTO Supplier(phone_number, shipment_time, company_name, type)
VALUES
   (1234567890, 3, 'ABC Suppliers', 'Food and Beverages'),
   (9876543210, 2, 'XYZ Distributors', 'Decorations'),
   (5551112222, 5, 'PQR Imports', 'Utensils'),
   (6043660789, 4, 'LMN Supplies', 'Ingredients'),
   (4443332222, 3, 'RST Exports', 'Festival Items'),
   (7785387823, 3, 'The Original Farm Goods', 'Ingredients')
   (7782267908, 3, 'Simon's Fresh Juice', 'Food and Beverages');

INSERT INTO Drinks_Provide (drink_name, phone_number, type)
VALUES
   ('Bubble Tea', 1234567890, 'Food and Beverages'),
   ('Iced Lemon Tea', 1234567890, 'Food and Beverages'),
   ('Mango Smoothie', 1234567890, 'Food and Beverages'),
   ('Green Tea Latte', 1234567890, 'Food and Beverages'),
   ('Watermelon Juice', 7782267908, 'Food and Beverages');


INSERT INTO Ingredients_Provide (ingredient_name, phone_number, type)
VALUES
('Chili Powder', 6043660789, 'Ingredients'),
('Soy Sauce', 6043660789, 'Ingredients'),
('Beef Slices', 6043660789, 'Ingredients'),
('Shrimp, 6043660789, 'Ingredients'),
```

('Carrots', 7785387823, 'Ingredients'),

```
INSERT INTO Process (ingredient_name, item_name)
VALUES
   ('chili powder', 'sour spicy soup'),
   ('carrots', 'sorted veggies'),
   ('soy sauce', 'egg fried rice'),
   ('beef slices', 'fatty beef'),
   ('shrimp', 'smashed shrimp');

INSERT INTO Dish_items (item_name, category, price)
VALUES ('sour spicy soup', cooked, 6)
        ('sorted veggies', raw, 8),
        ('egg fried rice', cooked, 5),
        ('fatty beef', cooked, 11),
        ('smashed shrimp', cooked, 11);

INSERT INTO Has_Sales (branch_number, item_name, date, amount, profit)
VALUES
  (8204, 'sour spicy soup', '2024-2-28', 80, 400),
  (3252, 'sorted veggies', '2024-2-28', 277, 1385),
  (3243, 'egg fried rice', '2024-2-28', 86, 172),
  (1332, 'fatty beef', '2024-2-28', 240, 1200),
  (1294, 'smashed shrimp', '2024-2-28', 280, 840)
```