University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 2

Date: 21 July 2024

Group Number: 06

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Brandan Bang	65585572	y0x6g	bbang@student.ubc.ca
Khushi Sharma	55400469	y2n6u	khushisharma.4804@gmail.com
Nikita Prabhu	24153306	m2s0i	nikip901@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.)

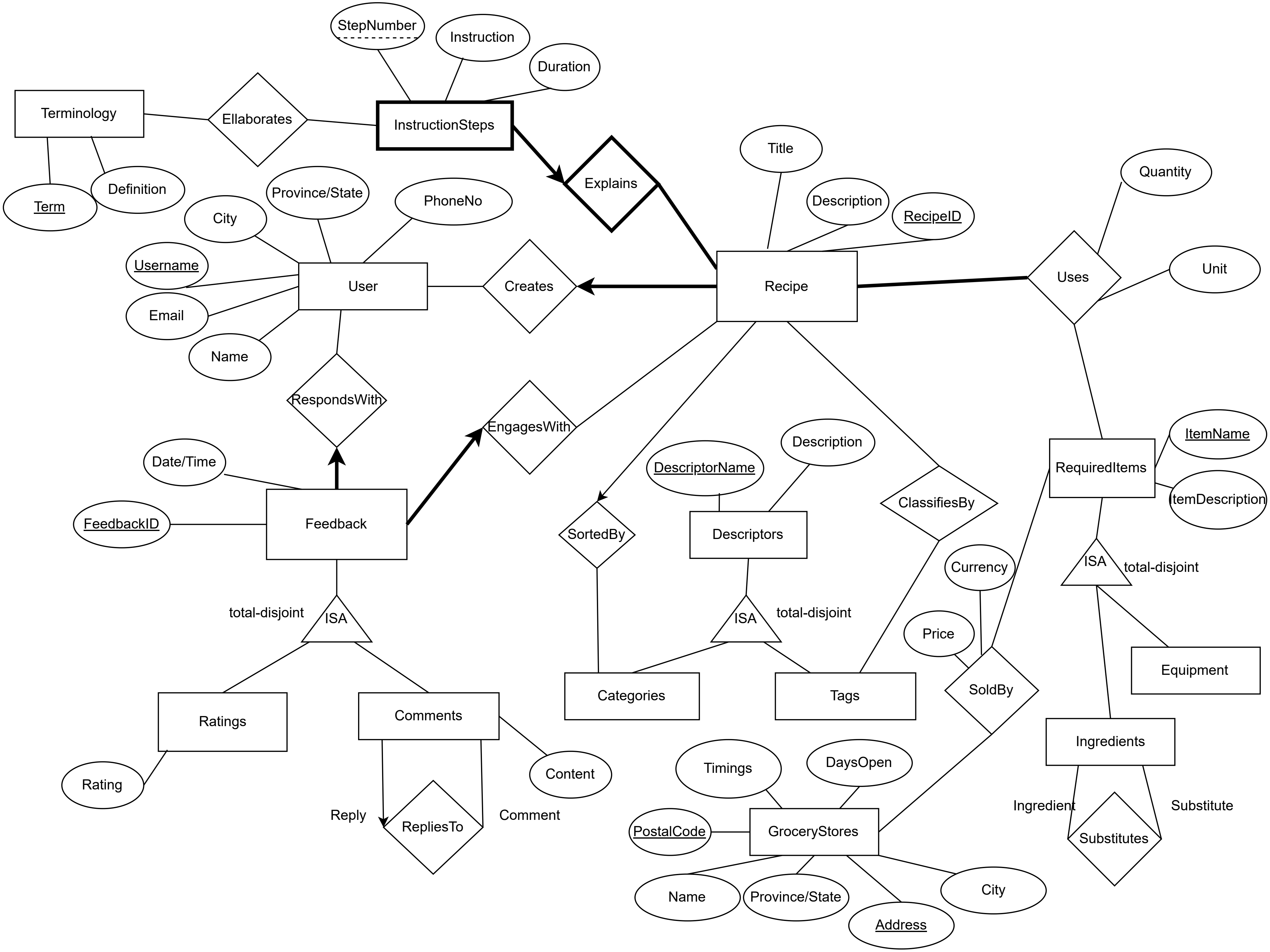
In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

2. Summary

Our primary purpose is to let people access one place where they can find everything they would need to cook their favorite dishes or even try to make new dishes. The recipes, the ingredients needed, substitutes, best places they can buy items required in the recipe, reviews about the recipe including other people's experiences with it. People can look for recipes based on the ingredients they have, cuisine preferences and dietary and time restrictions. Another use can be looking for the best nearby place to buy missing items. They can also rate and view other people's reviews of a particular recipe.

3. ER Diagram

We decided to incorporate our mentor's advice by adding the attribute postal code in the grocery store entity and making the Address and Postal Code the Primary Key. We realized that it is possible for the same chain of grocery stores to have locations on generic address locations (i.e. 123 Main St.). Note that Canada Post claims that Postal Code alone is not unique, so we still kept Address.
Other small changes include:
Renaming several generic attributes to clearly specify what it is referencing (i.e. Description → RecipeDescription)
Renaming the 'Name' attribute in Recipe to be 'Title'
Renaming 'Content' in Descriptors to 'DescriptionName'
Added 'Price' and 'Currency' attributes to 'SoldBy' relationship
Added 'PhoneNo' attribute to User
Added 'Timings' and 'DaysOpen' to GroceryStore


StepNumber
Instruction
Duration
Terminology
Ellaborates
InstructionSteps
Explains
Title
Quantity
Term
Definition
Province/State
PhoneNo
City
Description
RecipeID
Unit
Username
User
Creates
Recipe
Uses
Email
Name
RespondsWith
EngagesWith
ItemName
Date/Time
Description
DescriptorName
RequiredItems
ItemDescription
ClassifiesBy
FeedbackID
Feedback
SortedBy
Descriptors
Currency
ISA
total-disjoint
total-disjoint
ISA
ISA
total-disjoint
Price
Equipment
Categories
Tags
SoldBy
Ratings
Comments
Ingredients
Content
Rating
Timings
DaysOpen
Reply
RepliesTo
Comment
Ingredient
Substitute
Substitutes
PostalCode
GroceryStores
Name
Province/State
City
Address

4. Schema

InstructionSteps(StepNumber: INTEGER, Instruction: VARCHAR (300), **RecipeID:** INTEGER, Duration: DECIMAL)

Instruction NOT NULL

User(ProvinceState: VARCHAR(30), City: VARCHAR(60), Username: VARCHAR(60), Email: VARCHAR (320), Name: VARCHAR (40), PhoneNo: INTEGER)

Email NOT NULL
Email is UNIQUE
Email CANDIDATE KEY

GroceryStores(PostalCode: CHAR(6), StoreName: VARCHAR(50), ProvinceState: VARCHAR(60), Address: VARCHAR(100), City: VARCHAR(60), Timings: CHAR(11), DaysOpen: VARCHAR(30))

RequiredItems(ItemName: VARCHAR(50), ItemDescription: VARCHAR(300))

ISA Method 1: No point in having tables with just a single attribute

FeedbackRespondsWithEngagesWith(FeedbackID: INTEGER, DateTime: DATE, **RecipeID**: INTEGER, **Username** :VARCHAR(60))

ISA Method 2

CommentsRepliesTo(**FeedbackID**: INTEGER, Content: VARCHAR(300), **ParentID**: INTEGER)

Content NOT NULL

Rating (**FeedbackID**: INTEGER, Rating: INTEGER)

RATING NOT NULL

RecipeCreatesSortedBy(Title: VARCHAR(50),RecipeDescription: VARCHAR(200),RecipeID: INTEGER, **Username**: VARCHAR(60), **DescriptorName**: VARCHAR(50))

Descriptors(DescriptorName: VARCHAR(50), DescriptorDescription: VARCHAR(200))

ISA Method 1: No point in having tables with just a single attribute

Terminology(Term: VARCHAR(50), Definition: VARCHAR(300))
DEFINITION NOT NULL

Elaborates(**Term**: VARCHAR(50), **RecipeID:** INTEGER, **StepNumber**: INTEGER)

ClassifiesBy(**RecipeID:** INTEGER, **DescriptorName**: VARCHAR(50))

Uses(**RecipeID:** INTEGER, **ItemName:** VARCHAR(50), Quantity: DECIMAL, Unit**:** VARCHAR(15))

SoldBy(**<u>ItemName</u>**: VARCHAR(50), **<u>Address</u>**:VARCHAR(100), **<u>PostalCode</u>**: CHAR(6), Price: DECIMAL, Currency: CHAR(30)

Quantity NOT NULL
Units NOT NULL

Substitutes(**<u>IngredientName</u>**: VARCHAR(50), **<u>SubstituteName</u>**: VARCHAR(50))

TOTAL PARTICIPATION CONSTRAINTS:
All Recipes use ingredients
All Recipes have instruction steps
All Recipes has a user
All Feedback has a user
All feedback has an associated recipe

5. Functional Dependencies (FDs)

Descriptors:

- DescriptorName-> DescriptorDescription

InstructionSteps:

- RecipeID, StepNumber -> Instruction, Duration
- Instruction -> Duration

GroceryStores:

- PostalCode -> City, Province
- PostalCode, Address -> StoreName, City, Province
- StoreName -> Timings, DaysOpen

FeedbackRespondsWithEngagesWith:

- FeedbackID -> DateTime, RecipeID, Username
- DateTime, Username -> FeedbackID

CommentsRepliesTo:

- FeedbackID -> Content, ParentID

Rating:

- FeedbackID-> Rating

User:

- Username -> ProvinceState, City, Email, Name
- Email -> Username, Name
- Phone No -> ProvinceState, City

RequiredItems:

- ItemName->ItemDescription

RecipeCreatesSortedBy:

- Title, Username->RecipeID
- RecipeID -> Title, RecipeDescription, Username, DescriptorName

Terminology:

- Term->Definition

Substitutes:

- IngredientName -> SubstituteName
- SubstituteName -> IngredientName

SoldBy:

- PostalCode, Address, ItemName -> Price
- PostalCode -> Currency

Uses:

- RecipeID, ItemName -> Quantity, Unit

Elaborates:

- Trivial

ClassifiesBy:

- Trivial

6. Normalization (in BCNF)

Descriptors:

1. DescriptorName-> DescriptorDescription

   $DescriptorName^{+} = \{DescriptorName,\ DescriptorDescription\}$

   In BCNF; LHS contains superkey

   R(<u>DescriptorName</u> VARCHAR(50), DescriptorDescription VARCHAR(200))

InstructionSteps:

1. RecipeID, StepNumber -> Instruction, Duration
2. Instruction -> Duration

RecipeID, StepNumber	Instruction	Duration

- PK: RecipeID, StepNumber
- FK: RecipeID

$RecipeID,\ StepNumber^{+} = \{\,RecipeID,\ StepNumber,\ Instruction,\ Duration\}$

$Instruction^{+} = \{Instruction,\ Duration\}$

FD1 is in BCNF as the LHS is a superkey.
FD2 is not in BCNF as the LHS is not a superkey.

RecipeID, StepNumber || Instruction || Duration
R1(Instruction, Duration)
R2(RecipeID, StepNumber, Instruction)
<empty> || StepNumber, RecipeID || Instruction
R3(StepNumber, RecipeID, Instruction)
R4 (StepNumber, RecipeID)

R1(Instruction: VARCHAR(300), Duration: DECIMAL)
R3(StepNumber: INTEGER, **RecipeID**: INTEGER, **Instruction**: VARCHAR(300))

Note we did not include R4 as it is contained within R3

GroceryStores:

1. PostalCode -> City, ProvinceState
2. PostalCode, Address -> StoreName, City, ProvinceState, DaysOpen, Timings
3. StoreName -> DaysOpen, Timings

PostalCode, Address	StoreName	City, ProvinceState, Timings, DaysOpen

- PK: PostalCode, Address

$PostalCode^{+} = \{PostalCode,\ City,\ ProvinceState\}$

$PostalCode, Address^{+} = \{PostalCode,\ Address,\ StoreName,\ City,\ ProvinceState,\ DaysOpen,\ Timings$

$Name^{+} = \{StoreName,\ DaysOpen,\ Timings\}$

FD1 is not in BCNF as LHS is not a superkey
FD2 is in BCNF as LHS is a superkey
FD3 is not in BCNF as LHS is not a superkey

Timings, DaysOpen || PostalCode || City, ProvinceState
R1(PostalCode, City, ProvinceState)
R2(PostalCode, Address, StoreName, DaysOpen, Timings)

PostalCode, Address|| StoreName || DaysOpen, Timings
R3(StoreName, DaysOpen, Timings)
R4(StoreName, PostalCode, Address)
<blank> || PostalCode, Address || StoreName
R5(PostalCode, Address, StoreName)
R6(PostalCode, Address)

R1(PostalCode: CHAR(6), City: VARCHAR(60), ProvinceState: VARCHAR(60))
R3(StoreName: VARCHAR(50), DaysOpen: VARCHAR(30), Timings: CHAR(11))
R5(**PostalCode**: CHAR(6), Address: VARCHAR(100), **StoreName**: VARCHAR(50))

Note: we exclude R6 as it is contained within R5

FeedbackRespondsWithEngagesWith:

1. FeedbackID -> DateTime, RecipeID, Username
2. DateTime, Username -> FeedbackID

	FeedbackID, DateTime, Username	RecipeID

- PK: FeedbackID
- FK: RecipeID; Username
- CK: DateTime, Username

$FeedbackID^{+} = \{FeedbackID,\ DateTime,\ RecipeID,\ Username\}$
$DateTime,\ Username = \{DateTime,\ Username,\ FeedbackID,\ RecipeID\}$

FD1: In BCNF; LHS contains superkey
FD2: In BCNF; LHS contains superkey

R(FeedbackID INTEGER, DateTime: DATE, **RecipeID**: INTEGER, **Username**: VARCHAR(60))
CK: DateTime, Username

CommentsRepliesTo:

1. FeedbackID -> Content, ParentID

$FeedbackID^{+} = \{FeedbackID,\ Content,\ ParentID\}$

FD1 in BCNF; LHS contains superkey

R(**FeedbackID**: INTEGER, Content: VARCHAR(300), **ParentID**: INTEGER)

Rating:

1. FeedbackID-> Rating

$FeedbackID^{+} = \{FeedbackID,\ Rating\}$

FD1 in BCNF; LHS contains superkey

R(**FeedbackID**: INTEGER, Rating: INTEGER)

User:

1. Username -> ProvinceState, City, Email, Name, PhoneNo
2. Email -> Username, Name
3. Phone No -> ProvinceState, City

	Username, Email, PhoneNo	ProvinceState, City, Name

- PK: Username
- CK: Email

$Username^{+} = \{Username,\ ProvinceState,\ City,\ Email,\ Name,\ PhoneNo\}$

$Email^{+} = \{Email,\ Username,\ Name,\ ProvinceState,\ City,\ PhoneN\}$

$PhoneNo^{+} = \{PhoneNo,\ ProvinceState,\ City\}$

FD1 is in BCNF; LHS contains super key
FD2 is in BCNF; LHS contains super key
FD3 is not in BCNF; LHS is not a superkey

Username, Email, Name || PhoneNo || ProvinceState, City
R1(PhoneNo, ProvinceState, City)
R2(PhoneNo, Username, Email, Name)
<blank>|| Username || PhoneNo, Email, Name
R3(Username, PhoneNo, Email, Name)
R4(Username)

R1(PhoneNo: INTEGER, ProvinceState: VARCHAR(30), City: VARCHAR(60))
R3(Username: VARCHAR(60), **PhoneNo**: INTEGER, Email: VARCHAR(320), Name: VARCHAR(40))
CK: Email

Note: we exclude R4 as it is contained within R3

RequiredItems:

1. ItemName->ItemDescription

$Name^{+} = \{ItemName,\ ItemDescription\}$

FD1 in BCNF; LHS contains superkey

R(ItemName: VARCHAR(50), ItemDescription: VARCHAR(300))

RecipeCreatesSortedBy:

1. Title, Username->RecipeID
2. RecipeID -> Title, RecipeDescription, Username, DescriptorName

	RecipeID, Title, Username	RecipeDescription, DescriptorName

- PK: RecipeID
- FK: Username; DescriptorName

$Title,\ Username^{+}\ = \{Title,\ Username,\ RecipeID,\ RecipeDescription,\ DescriptorName\}$

$RecipeID^{+}\ = \{RecipeID,\ Title,\ RecipeDescription,\ Username,\ DescriptorName\}$

FD1 is in BCNF; LHS contains superkey
FD2 is in BCNF; LHS contains superkey

R(Title: VARCHAR(50),RecipeDescription: VARCHAR(200),RecipeID: INTEGER, **Username**: VARCHAR(60), **DescriptorName**: VARCHAR(50))
CK: Title, Username

Terminology:

1. Term->Definition

$Term^{+}\ = \{Term,\ Definition\}$

FD1 in BCNF; LHS contains superkey

R(Term: VARCHAR(50), Definition: VARCHAR(100))

Substitutes:

1. IngredientName -> SubstituteName
2. SubstituteName -> IngredientName

	IngredientName, SubstituteName	

- PK: IngredientName
- CK: SubstituteName

$IngredientName^{+} = \{IngredientName\ , SubstituteName\ \}$

$SubstituteName^{+} = \{SubstituteName\ \ , IngredientName\}$

FD1 is in BCNF; LHS contains superkey
FD2 is in BCNF; LHS contains superkey

R(**IngredientName**: VARCHAR(50), **SubstituteName**: VARCHAR(50))
CK: SubstituteName

SoldBy:

1. PostalCode, Address, ItemName -> Price
2. PostalCode -> Currency

PostalCode, Address, ItemName		Price, Currency

- PK: PostalCode, Address, ItemName
- FK: ItemName

$PostalCode,\ Address,\ ItemName^{+} = \{PostalCode,\ Address,\ ItemName,\ Price,\ Currency\}$

$PostalCode^{+} = \{PostalCode,\ Currency\}$

FD1 in BCNF; LHS contains superkey
FD2 not in BCND; LHS is not a superkey

Address, ItemName, Price || PostalCode || Currency
R1(PostalCode, Currency)
R2(PostalCode, Address, ItemName, Price)
<blank> || PostalCode, Address, ItemName || Price
R3(PostalCode, Address, ItemName, Price)
R4(PostalCode, Address, ItemName)

R1(**PostalCode**: CHAR(6), Currency: CHAR(30))
R3(**PostalCode**: CHAR(6), Address: VARCHAR(100), **ItemName**: VARCHAR(50), Price: DECIMAL)

Note: we exclude R4 as it is contained within R3

Uses:

1. RecipeID, ItemName -> Quantity, Unit

$RecipeID,\ ItemName^{+} = \{RecipeID,\ ItemName, Quantity,\ Unit\}$

FD1 in BCNF; LHS contains superkey

R(**RecipeID:** INTEGER, **ItemName:** VARCHAR(50), Quantity: DECIMAL, Unit**:** VARCHAR(15))

Elaborates:

- Only Trivial Case; no Decomposition Needed

ClassifiesBy:

- Only Trivial Case; no Decomposition Needed

7. The SQL DDL statements required to create all the tables

```
CREATE TABLE UserLocation
(PhoneNo INTEGER PRIMARY KEY,
ProvinceState  VARCHAR(30),
City VARCHAR(60));

CREATE TABLE UserDetails
(Username VARCHAR(60) NOT NULL UNIQUE,
PhoneNo INTEGER,
Email VARCHAR (320) NOT NULL UNIQUE,
Name VARCHAR (40),
PRIMARY KEY(PhoneNo),
FOREIGN KEY (PhoneNo) REFERENCES UserLocation(PhoneNo) ON DELETE CASCADE
ON UPDATE CASCADE);

CREATE TABLE Descriptors
(DescriptorName VARCHAR(50) PRIMARY KEY,
DescriptorDescription VARCHAR(200));

CREATE TABLE RecipeCreatesSortedBy (
   Title VARCHAR(50),
   RecipeDescription VARCHAR(200),
   RecipeID INTEGER PRIMARY KEY,
   Username VARCHAR(60),
   DescriptorName VARCHAR(50),
   FOREIGN KEY (Username) REFERENCES UserDetails(Username) ON DELETE
CASCADE,
   FOREIGN KEY (DescriptorName) REFERENCES Descriptors(DescriptorName) ON DELETE
SET NULL ON UPDATE CASCADE
);
CREATE TABLE InstructionTime (
         Instruction VARCHAR (300) PRIMARY KEY,
         Duration DECIMAL
);
```

```
CREATE TABLE InstructionStep (
        RecipeID INTEGER,
        StepNumber INTEGER,
        Instruction VARCHAR (300),
        PRIMARY KEY (RecipeID, StepNumber),
        FOREIGN KEY (RecipeID) REFERENCES RecipeCreatesSortedBy(RecipeID) ON
DELETE CASCADE ON UPDATE CASCADE,
        FOREIGN KEY (Instruction) REFERENCES InstructionTime(Instruction) ON    DELETE
CASCADE  ON UPDATE CASCADE
);

CREATE TABLE GroceryStoresArea
(PostalCode CHAR(6) PRIMARY KEY,
City VARCHAR(60),
ProvinceState VARCHAR(60));

CREATE TABLE GroceryStoreTimings
(StoreName VARCHAR(50) PRIMARY KEY,
DaysOpen VARCHAR(30),
Timings CHAR(11));

CREATE TABLE GroceryStoreLocation
(PostalCode CHAR(6),
 Address VARCHAR(100),
StoreName VARCHAR(50),
PRIMARY KEY(PostalCode,Address),
FOREIGN KEY (PostalCode) REFERENCES GroceryStoresArea(PostalCode) ON DELETE
CASCADE  ON UPDATE CASCADE,
FOREIGN KEY (StoreName) REFERENCES GroceryStoreTimings(StoreName) ON DELETE
CASCADE  ON UPDATE CASCADE );

CREATE TABLE FeedbackRespondsWithEngagesWith
(FeedbackID INTEGER  PRIMARY KEY,
 DateTime DATE,
 RecipeID INTEGER,
 Username VARCHAR(60),
FOREIGN KEY (RecipeID) REFERENCES RecipeCreatesSortedBy(RecipeID) ON DELETE
CASCADE,
FOREIGN KEY (Username) REFERENCES UserDetails(Username) ON UPDATE NO
ACTION);
```

```
CREATE TABLE CommentsRepliesTo
(FeedbackID INTEGER,
 Content VARCHAR(300) NOT NULL,
 ParentID INTEGER ,
 PRIMARY KEY(FeedbackID),
FOREIGN KEY (FeedbackID) REFERENCES
FeedbackRespondsWithEngagesWith(FeedbackID) ON DELETE CASCADE,
FOREIGN KEY (ParentID) REFERENCES CommentsRepliesTo(FeedbackID) ON DELETE NO
ACTION);

CREATE TABLE Rating
(FeedbackID INTEGER PRIMARY KEY,
Rating INTEGER NOT NULL,
FOREIGN KEY (FeedbackID) REFERENCES
FeedbackRespondsWithEngagesWith(FeedbackID) ON DELETE CASCADE);

CREATE TABLE RequiredItems (
   ItemName VARCHAR(50) PRIMARY KEY,
   ItemDescription VARCHAR(300)
);

CREATE TABLE Terminology (
   Term VARCHAR(50) PRIMARY KEY,
   Definition VARCHAR(300) NOT NULL
);

CREATE TABLE Substitutes(
   IngredientName VARCHAR(50),
   SubstituteName VARCHAR(50),
   PRIMARY KEY (IngredientName, SubstituteName),
   FOREIGN KEY (IngredientName) REFERENCES RequiredItems(ItemName) ON DELETE
NO ACTION,
   FOREIGN KEY (SubstituteName) REFERENCES RequiredItems(ItemName) ON DELETE
NO ACTION
)

CREATE TABLE SoldByCurrency(
    PostalCode CHAR(6) PRIMARY KEY,
    Currency CHAR(30),
    FOREIGN KEY (PostalCode) REFERENCES GroceryStoresArea(PostalCode) ON DELETE
CASCADE ON UPDATE CASCADE
)
```

```
CREATE TABLE SoldByLocation (
   PostalCode CHAR(6),
   Address VARCHAR(100),
   ItemName VARCHAR(50),
   Price DECIMAL,
   PRIMARY KEY (PostalCode, Address, ItemName),

   FOREIGN KEY (PostalCode,Address) REFERENCES
GroceryStoreLocation(PostalCode,Address) ON DELETE CASCADE ON UPDATE CASCADE,
   FOREIGN KEY (ItemName) REFERENCES RequiredItems(ItemName) ON DELETE
CASCADE ON UPDATE CASCADE
);

CREATE TABLE  Uses (
   RecipeID INTEGER,
   ItemName VARCHAR(50),
   Quantity DECIMAL NOT NULL,
   Unit VARCHAR(15),
   PRIMARY KEY (RecipeID, ItemName),
   FOREIGN KEY (RecipeID) REFERENCES RecipeCreatesSortedBy(RecipeID) ON
DELETE CASCADE,
   FOREIGN KEY (ItemName) REFERENCES RequiredItems(ItemName) ON DELETE
CASCADE ON UPDATE CASCADE
);

CREATE TABLE Elaborates (
   Term VARCHAR(50),
   RecipeID INTEGER,
   StepNumber INTEGER,
   PRIMARY KEY (Term, RecipeID, StepNumber),
   FOREIGN KEY (Term) REFERENCES Terminology(Term) ON DELETE CASCADE ON
UPDATE CASCADE,
   FOREIGN KEY (RecipeID, StepNumber) REFERENCES InstructionStep(RecipeID,
StepNumber) ON DELETE CASCADE);

CREATE TABLE ClassifiesBy (
   RecipeID INTEGER,
   DescriptorName VARCHAR(50),
   PRIMARY KEY (RecipeID, DescriptorName),
   FOREIGN KEY (RecipeID) REFERENCES RecipeCreatesSortedBy(RecipeID) ON DELETE
CASCADE,
   FOREIGN KEY (DescriptorName) REFERENCES Descriptors(DescriptorName) ON DELETE
CASCADE ON UPDATE CASCADE
);
```

8. INSERT statements to populate each table with at least 5 tuples.

```
INSERT INTO UserLocation (PhoneNo,ProvinceState, City)
VALUES
(1234567890,'British Columbia','Vancouver'),
(1345678901,'Zinj','Manama'),
(1456789012,'Quebec', 'Montreal'),
(1567890123,'Calgary','Alberta'),
(1678901234,'Manitoba', 'Winnipeg'),
(1678901284,'Mumbai', 'Maharashtra');

INSERT INTO UserDetails ( Username,PhoneNo,Email,Name)
VALUES
('chef_janes',1234567890, 'jane.stevens@gmail.com','Jane Stevens'),
('greatestBossEver',1345678901,'micheal.scott@yahoo.com', 'Michael Scott'),
('culinary_queen248',1456789012,'alice.cooper_12@gmail.com', 'Alice Cooper'),
('bake_master',1567890123,'james.brown@gmail.com','James Brown'),
('spicylover',1678901234,'gordon.ramsey@yahoo.com', 'Gordon Ramsey'),
('best_cook_2',1678901284, 'srk_02@gmail.com', 'Shah Rukh Khan');

INSERT INTO Descriptors(DescriptorName, DescriptorDescription)
VALUES ('pescatarian', 'a person who eats fish but no other meat'),
('lunch', 'a meal typically eaten at the middle of the day'),
('French Recipe', 'A recipe that belongs to the French cuisine'),
('Spicy', 'a dish flavoured with spices or chilli peppers'),
('quick', 'a meal that takes less than 10 minutes to prepare'),
('dessert', 'The sweet course eaten at the end of a meal');

INSERT INTO RecipeCreatesSortedBy(Title ,RecipeDescription, RecipeID, Username,
DescriptorName)
VALUES
('Classic Chocolate Cake', 'A rich and moist chocolate cake perfect for any occasion.', 101,
'chef_janes', 'dessert'),
('Spaghetti Bolognese', 'A traditional Italian pasta dish with a flavorful meat sauce.', 103,
'best_cook_2', 'lunch'),
('Vegan Tacos', 'Delicious and healthy vegan tacos packed with fresh vegetables.', 102,
'culinary_queen248', 'spicy'),
('Lemon Drizzle Cake', 'A zesty and moist lemon cake with a tangy drizzle topping.', 105,
'bake_master', 'dessert'),
('Spicy Chicken Curry', 'A hearty and spicy chicken curry with a rich, flavorful sauce.', 104,
'spicylover', 'spicy'),
('Avocado Toast', 'Simple and delicious avocado toast, perfect for a quick breakfast.', 106,
'chef_janes', 'quick');
```

```
INSERT INTO InstructionTime (Instruction, Duration)
VALUES
('Preheat the oven to 350 degrees F.',15),
( 'Mix flour and sugar in a bowl.',10),
( 'Chop the onions into julienne strips and dice the garlic.',15),
( 'Heat oil in a pan and sauté onions and garlic. Then deglaze the pan with white wine.', 20),
( 'Boil the pasta to al dente.',10),
( 'Blanch the spinach.',5),
('Toast the bread.',3),
('Mix pasta with prepared sauce.',1);

INSERT INTO InstructionStep (RecipeID,Instruction, StepNumber)
VALUES
(101, 'Preheat the oven to 350 degrees F.', 1),
(101, 'Mix flour and sugar in a bowl.', 2),
(102, 'Chop the onions into julienne strips and dice the garlic.', 1),
(102, 'Heat oil in a pan and sauté onions and garlic. Then deglaze the pan with white wine.', 2),
(103, 'Boil the pasta to al dente.', 1),
(103, 'Mix pasta with prepared sauce.', 2),
(104, 'Blanch the spinach.', 1),
(106, 'Toast the bread.', 1);

INSERT INTO GroceryStoresArea (PostalCode, City, ProvinceState)
VALUES
('V5K0A1','Vancouver','British Columbia'),
('M5V2T6','Toronto', 'Ontario'),
('H2Z1J9', 'Montreal', 'Quebec'),
('T2P3G7','Calgary', 'Alberta'),
('R3B0R5', 'Winnipeg', 'Manitoba');

INSERT INTO GroceryStoreTimings (StoreName, DaysOpen, Timings)
VALUES
('Save On Foods','Mon-Sun','08:00-23:00'),
('Loblaws','Mon-Sun','09:00-22:00'),
('PC Express','Mon-Sat','07:00-21:00'),
('Safeway', 'Mon-Sun','10:00-23:00'),
('No Frills','Mon-Fri', '08:00-23:00');
```

```
INSERT INTO GroceryStoreLocation (PostalCode,Address,StoreName)
VALUES
('V5K0A1','1234 Main St','Save On Foods'),
('M5V2T6', '5678 Queen St', 'Loblaws'),
('H2Z1J9','9101 Rue St.','Save On Foods'),
('T2P3G7','1213 4th Ave','Safeway'),
('R3B0R5','1415 Ellice Ave', 'No Frills');

INSERT INTO RequiredItems(ItemName, ItemDescription)
VALUES
('Flour', 'All-purpose flour used for baking and cooking.'),
('Sugar', 'Granulated white sugar for sweetening.'),
('Eggs', 'Fresh eggs for baking and cooking.'),
('Whisk', 'A utensil for whipping eggs or cream'),
('Baking Tray', 'A metal tray on which food may be cooked in an oven'),
('White Wine','Wine made from grapes without using their skin'),
('Lemon','A yellow citrus fruit having a thick skin'),
('Chicken','(Here) Meat obtained from the domestic animal chicken'),
('Tofu','A type of curd made from soybeans'),
('Chicken Broth','Broth obtained from boiling chicken peices'),
('Vegetable Broth','Broth obtained from boiling vegetables'),
('Lime','A green citrus fruit'),
('Sour Cream','Cream obtained by fermenting it with bacteria'),
('Greek Yogurt','Yogurt that has been strained');

INSERT INTO FeedbackRespondsWithEngagesWith (FeedbackID, DateTime, RecipeID,
Username)
VALUES
(1, '2024-07-19 14:30:00', '101', 'chef_janes'),
(2, '2024-07-19 15:00:00', '102', 'best_cook_2'),
(3, '2024-07-20 09:15:00', '101', 'culinary_queen248'),
(4, '2024-07-21 10:45:00', '103', 'bake_master'),
(5, '2024-07-20 11:30:00', '101', 'spicylover'),
(6,'2024-07-20 11:30:00','102','spicylover'),
(7,'2024-07-20 11:30:00','101','bake_master');
```

```
INSERT INTO CommentsRepliesTo (FeedbackID, Content, ParentID)
VALUES
(1, 'This recipe is fantastic! The instructions were clear and easy to follow.', NULL),
(2, 'I loved the flavours in this dish. Will definitely make it again!', NULL),
(3, 'Great recipe! I added some extra spices and it turned out perfect :)', NULL),
(4, 'Thanks for sharing this recipe! It was a hit at my dinner party', NULL),
(5, 'I agree with you, the instructions were very clear unfortunately didn’t like it as much as I
thought I would.', 1),
(6, 'I had the same experience! The flavours were amazing!!!', 2),
(7, 'What spices did you add?? I’d love to try that next time.', 3);

INSERT INTO Rating (FeedbackID, Rating)
VALUES
(1, 5),
(2, 4),
(3, 5),
(4, 4),
(5, 3),
(6, 4);

INSERT INTO Terminology(Term,Definition)
VALUES ('Al dente', 'This refers to cooking something,usually pasta or rice, such that it is firm.'),
('Blanch', 'This is a method of cooking food where it is immersed in boiling water for a short
period then put in an ice bath.'),
('Julienne', 'A culinary cutting method such that the food is cut into long strips resembling a
matchstick.'),
('Dice', 'A culinary cutting method such that the food is cut into small squares.'),
('Deglaze', 'When bits of food stuck to the pan is released using liquids(for example, wine).');

INSERT INTO Elaborates(Term,RecipeID,StepNumber)
VALUES ('Al dente',103,1),
('Blanch',104,1),
('Deglaze',102,2),
('Julienne',102,1),
('Dice',102,1);

INSERT INTO ClassifiesBy(RecipeID,DescriptorName)
VALUES(101,'dessert'),
(103,'lunch'),
(102,'spicy'),
(104,'spicy'),
(106,'quick');
```

```
INSERT INTO Uses(RecipeID,ItemName,Quantity,Unit)
VALUES(101, 'Whisk', 1, NULL),
(103, 'Eggs', 0.5, 'kg'),
(104, 'Flour', 400, 'grams'),
(106, 'Eggs', 1, 'unit'),
(101, 'Sugar', 0.2, 'kg');


INSERT INTO SoldByCurrency(PostalCode,Currency)
VALUES('V5K0A1','CAD'),
('M5V2T6','CAD'),
('H2Z1J9','CAD'),
('R3B0R5','CAD'),
('T2P3G7','USD');


INSERT INTO SoldByLocation( PostalCode,Address,ItemName,Price)
VALUES ('V5K0A1','1234 Main St','Whisk',4),
('M5V2T6','5678 Queen St','Sugar',0.7),
('T2P3G7','1213 4th Ave','Flour',0.9),
('T2P3G7','1213 4th Ave','Eggs',0.8),
('R3B0R5','1415 Ellice Ave','Baking Tray',3),
('T2P3G7','1213 4th Ave','Baking Tray',4);

INSERT INTO Substitutes(IngredientName, SubstituteName)
VALUES ('White Wine','Lemon'),
('Chicken','Tofu'),
('Chicken Broth','Vegetable Broth'),
('Lemon','Lime'),
('Sour Cream','Greek Yogurt');
```