

CPSC 304 Project Cover Page

Milestone #: 3

Date: March 6, 2024

Group Number: 9

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Jacky Feng	45105616	k4a1k	Jfeng14@student.ubc.ca
Safiullah Kaleem	57876609	s9r9w	skaleem@student.ubc.ca
Mohammad Fakhir	11939493	i2r8j	mfakhir@student.ubc.ca

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

[Link to the project](#)

SQL DDL and Insert Statements:

```
<?php

function dropTables($connection) {
    $dropTables = "
        DROP TABLE IF EXISTS Reviews CASCADE;
        DROP TABLE IF EXISTS Posts CASCADE;
        DROP TABLE IF EXISTS Utilizes CASCADE;
        DROP TABLE IF EXISTS UserRanking CASCADE;
        DROP TABLE IF EXISTS Participates CASCADE;
        DROP TABLE IF EXISTS Follows CASCADE;
        DROP TABLE IF EXISTS Contains CASCADE;
        DROP TABLE IF EXISTS Ingredient CASCADE;
        DROP TABLE IF EXISTS Video CASCADE;
        DROP TABLE IF EXISTS Recipe CASCADE;
        DROP TABLE IF EXISTS RecipeDetails CASCADE;
        DROP TABLE IF EXISTS CookingEquipmentName CASCADE;
        DROP TABLE IF EXISTS CookingEquipment CASCADE;
        DROP TABLE IF EXISTS Review CASCADE;
        DROP TABLE IF EXISTS ReviewDetails CASCADE;
        DROP TABLE IF EXISTS EventDetails CASCADE;
        DROP TABLE IF EXISTS EventLocation CASCADE;
        DROP TABLE IF EXISTS Leaderboard CASCADE;
        DROP TABLE IF EXISTS ProfessionalChef CASCADE;
        DROP TABLE IF EXISTS ProfessionalChefSkill CASCADE;
        DROP TABLE IF EXISTS HomeCook CASCADE;
        DROP TABLE IF EXISTS AppUser CASCADE;
        DROP TABLE IF EXISTS UserDetails CASCADE;
    ";

    try {
        $connection->beginTransaction();
        $connection->exec($dropTables);
        $connection->commit();
        echo "<p>All tables dropped successfully.</p>\n";
    }
}
```

University of British Columbia, Vancouver

Department of Computer Science

```
} catch (PDOException $e) {
    $connection->rollBack();
    die("Error dropping tables: " . $e->getMessage());
}

}

function createTables($connection){
    $intilizeTables =

    "CREATE TABLE IF NOT EXISTS UserDetails (
        NumberOfFollowers INTEGER DEFAULT 0,
        NumberOfFollowing INTEGER DEFAULT 0,
        Age INTEGER NOT NULL,
        Username VARCHAR(30),
        Password VARCHAR(255),
        PRIMARY KEY (Username, Password)
    );

    CREATE TABLE IF NOT EXISTS AppUser (
        Username VARCHAR (30) NOT NULL UNIQUE,
        Password VARCHAR (255) NOT NULL,
        UserID INTEGER,
        PRIMARY KEY (UserID),
        FOREIGN KEY (UserName, Password) REFERENCES UserDetails(UserName,
Password)
        ON DELETE CASCADE
        ON UPDATE CASCADE
    );

    CREATE TABLE IF NOT EXISTS HomeCook (
        FavouriteCuisine VARCHAR (30),
        HobbyistLevel VARCHAR(30) DEFAULT 'Amateur',
        UserID INTEGER,
        PRIMARY KEY (UserID),
        FOREIGN KEY (UserID) REFERENCES AppUser
        ON DELETE CASCADE ON UPDATE CASCADE
    );

    CREATE TABLE IF NOT EXISTS ProfessionalChefSkill (
        RestaurantAffiliation VARCHAR (30),
        RestaurantLocation VARCHAR (30),
```

University of British Columbia, Vancouver

Department of Computer Science

```
Certification VARCHAR (30),
PRIMARY KEY (RestaurantAffiliation, RestaurantLocation)
);

CREATE TABLE IF NOT EXISTS ProfessionalChef (
    RestaurantAffiliation VARCHAR (30),
    RestaurantLocation VARCHAR (30),
    UserID INTEGER,
    PRIMARY KEY (UserID),
    FOREIGN KEY (UserID) REFERENCES AppUser
    ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (RestaurantAffiliation, RestaurantLocation)
REFERENCES ProfessionalChefSkill
    ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE IF NOT EXISTS Leaderboard (
    LeaderboardCategory VARCHAR (30) NOT NULL UNIQUE,
    Prize VARCHAR (30),
    PRIMARY KEY (LeaderboardCategory)
);

CREATE TABLE IF NOT EXISTS EventLocation (
    Category VARCHAR (30),
    EntryFee INTEGER,
    Location VARCHAR (30) NOT NULL,
    PRIMARY KEY (Category, EntryFee)
);

CREATE TABLE IF NOT EXISTS EventDetails (
    Category VARCHAR (30) NOT NULL,
    EntryFee INTEGER,
    EventID VARCHAR (30),
    Date VARCHAR (30),
    PRIMARY KEY (EventID),
    FOREIGN KEY (Category, EntryFee) REFERENCES EventLocation ON
DELETE CASCADE ON UPDATE CASCADE
);
```

University of British Columbia, Vancouver

Department of Computer Science

```
CREATE TABLE IF NOT EXISTS ReviewDetails (  
    TimePosted DATE,  
    Comment VARCHAR (30),  
    Rating INTEGER DEFAULT 0,  
    PRIMARY KEY (TimePosted, Comment)  
);  
  
CREATE TABLE IF NOT EXISTS Review (  
    TimePosted DATE NOT NULL,  
    Comment VARCHAR (30),  
    ReviewID INTEGER,  
    PRIMARY KEY (ReviewID),  
    FOREIGN KEY (TimePosted, Comment) REFERENCES ReviewDetails  
    ON DELETE CASCADE  
    ON UPDATE CASCADE  
);  
  
CREATE TABLE IF NOT EXISTS CookingEquipment (  
    Price INTEGER ,  
    Category VARCHAR (30),  
    Quality VARCHAR (30),  
    Brand VARCHAR (20) NOT NULL,  
    PRIMARY KEY (Price, Category, Quality)  
);  
  
CREATE TABLE IF NOT EXISTS CookingEquipmentName (  
    Name VARCHAR (30),  
    Price INTEGER ,  
    Category VARCHAR (30),  
    Quality VARCHAR (30),  
    PRIMARY KEY (Name, Price, Category, Quality),  
    FOREIGN KEY (Price, Category, Quality) REFERENCES  
CookingEquipment  
    ON DELETE CASCADE  
    ON UPDATE CASCADE  
);  
  
CREATE TABLE IF NOT EXISTS RecipeDetails (  
    PublishDate DATE,  
    Title VARCHAR (20),  
    Description VARCHAR (200),
```

```
        Culture VARCHAR (20) NOT NULL,
        Difficulty VARCHAR (20) NOT NULL,
        Serving INTEGER,
        PRIMARY KEY (PublishDate, Title)
    );

CREATE TABLE IF NOT EXISTS Recipe (
    RecipeID INTEGER ,
    PublishDate DATE NOT NULL,
    EstimatedTime INTEGER NOT NULL,
    Title VARCHAR (20) NOT NULL,
    PRIMARY KEY (RecipeID),
    FOREIGN KEY (PublishDate, Title) REFERENCES RecipeDetails
    ON DELETE CASCADE
    ON UPDATE CASCADE
);

CREATE TABLE IF NOT EXISTS Video (
    Name VARCHAR (30),
    UploadTime DATE,
    RecipeID INTEGER,
    VideoURL VARCHAR (100),
    Duration INTEGER NOT NULL,
    Views VARCHAR (30) DEFAULT 0,
    PRIMARY KEY (Name, UploadTime, RecipeID),
    FOREIGN KEY (RecipeID) REFERENCES Recipe
    ON DELETE CASCADE
    ON UPDATE CASCADE
);

CREATE TABLE IF NOT EXISTS Ingredient (
    Name VARCHAR (30),
    AllergenInfo VARCHAR (30) DEFAULT 'None',
    PRIMARY KEY (Name)
);

CREATE TABLE IF NOT EXISTS Contains (
    RecipeID INTEGER,
    Name VARCHAR (30),
    PRIMARY KEY (RecipeID, Name),
    FOREIGN KEY (RecipeID) REFERENCES Recipe
    ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (Name) REFERENCES Ingredient
    ON DELETE CASCADE ON UPDATE CASCADE
);
```

```
);

CREATE TABLE IF NOT EXISTS Follows (
    UserID1 INTEGER,
    UserID2 INTEGER,
    PRIMARY KEY (UserID1, UserID2),
    FOREIGN KEY (UserID1) REFERENCES AppUser
    ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (UserID2) REFERENCES AppUser
    ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE IF NOT EXISTS Participates (
    UserID INTEGER ,
    EventID VARCHAR(30),
    PRIMARY KEY (UserID, EventID),
    FOREIGN KEY (UserID) REFERENCES AppUser
    ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (EventID) REFERENCES EventDetails
    ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE IF NOT EXISTS UserRanking (
    UserID INTEGER,
    LeaderboardCategory VARCHAR (30),
    Points INTEGER,
    Position INTEGER,
    PRIMARY KEY (UserID, LeaderboardCategory),
    FOREIGN KEY (UserID) REFERENCES AppUser
    ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (LeaderboardCategory) REFERENCES Leaderboard
    ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE IF NOT EXISTS Utilizes (
    Name VARCHAR (30),
    Price INTEGER ,
    Category VARCHAR (30),
    Quality VARCHAR (30),

    RecipeID INTEGER,
    PRIMARY KEY (RecipeID, Name, Price, Category, Quality),
    FOREIGN KEY (RecipeID) REFERENCES Recipe
    ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (Name, Price, Category, Quality) REFERENCES
CookingEquipmentName
```

```
        ON DELETE CASCADE
        ON UPDATE CASCADE

    );

    CREATE TABLE IF NOT EXISTS Posts (
        RecipeID INTEGER,
        UserID INTEGER,
        PRIMARY KEY (RecipeID, UserID),
        FOREIGN KEY (RecipeID) REFERENCES Recipe
            ON DELETE CASCADE ON UPDATE CASCADE,
        FOREIGN KEY (UserID) REFERENCES AppUser
            ON DELETE CASCADE ON UPDATE CASCADE
    );

    CREATE TABLE IF NOT EXISTS Reviews (
        RecipeID INTEGER,
        ReviewID INTEGER,
        UserID INTEGER,
        PRIMARY KEY (ReviewID, UserID, RecipeID),
        FOREIGN KEY (ReviewID) REFERENCES Review
            ON DELETE CASCADE ON UPDATE CASCADE,
        FOREIGN KEY (UserID) REFERENCES AppUser
            ON DELETE CASCADE ON UPDATE CASCADE,
        FOREIGN KEY (RecipeID) REFERENCES Recipe
            ON DELETE CASCADE ON UPDATE CASCADE
    );
";

$connection->beginTransaction();
$connection->exec($intilizetables);

$connection->commit();

}

function sampleData($connection){
$insertQuery = "INSERT INTO UserDetails (NumberOfFollowers,
NumberOfFollowing, Age, Username, Password)
VALUES
(100, 50, 25, 'jacky', '110'),
(200, 75, 30, 'safiullah', '320'),
(150, 60, 28, 'mohammad', '310'),
(120, 45, 22, 'seva', '221'),
```


University of British Columbia, Vancouver

Department of Computer Science

```
(80, 35, 27, 'raymond', '304'),  
(145, 90, 89, 'nafis', '304'),  
(89, 453, 43, 'gregor', '110'),  
(57, 329, 71, 'clune', '340'),  
(67, 78, 56, 'feeley', '213'),  
(98, 46, 32, 'holmes', '310'),  
(87, 20, 102, 'pottinger', '304'),  
(68, 198, 125, 'wolfman', '313'),  
(56, 980, 98, 'jordon', '213'),  
(110, 589, 75, 'geoffrey', '121'),  
(120, 437, 32, 'bradley', '320'),  
(89, 84938, 43, 'cinda', '221'),  
(96, 473827483, 87, 'carter', '210'),  
(48, 2490, 12, 'patrice', '313'),  
(82, 437843, 35, 'ratna', '295'),  
(102, 3848, 23, 'anton', '999')";
```

```
$insertQuery2 = "INSERT INTO AppUser (Username, Password, UserID)  
VALUES
```

```
('jacky', '110', 1),  
( 'safiullah', '320', 2),  
( 'mohammad', '310', 3),  
( 'seva', '221', 4),  
( 'raymond', '304', 5),  
( 'nafis', '304', 6),  
( 'gregor', '110', 7),  
( 'clune', '340', 8),  
( 'feeley', '213', 9),  
( 'holmes', '310', 10),  
( 'pottinger', '304', 11),  
( 'wolfman', '313', 12),  
( 'jordon', '213', 13),  
( 'geoffrey', '121', 14),  
( 'bradley', '320', 15),  
( 'cinda', '221', 16),  
( 'carter', '210', 17),  
( 'patrice', '313', 18),  
( 'ratna', '295', 19),  
( 'anton', '999', 20)";
```

```
$insertQuery5 = "INSERT INTO HomeCook (FavouriteCuisine, HobbyistLevel,  
UserID)
```

```
VALUES
```

```
('Italian', 'Amateur', 1),
```

University of British Columbia, Vancouver

Department of Computer Science

```
('Mexican', 'Intermediate', 2),
('Japanese', 'Advanced', 3),
('Italian', 'Amateur', 4),
('British', 'Advanced', 5)";

$insertQuery7 = "INSERT INTO ProfessionalChefSkill (RestaurantAffiliation,
RestaurantLocation, Certification)
VALUES
('Restaurant1', 'Location1', 'Chef Certification 1'),
('Restaurant2', 'Location2', 'Chef Certification 2'),
('Restaurant3', 'Location3', 'Chef Certification 3'),
('Restaurant4', 'Location4', 'Chef Certification 4'),
('Restaurant5', 'Location5', 'Chef Certification 5')";

$insertQuery6 = "INSERT INTO ProfessionalChef (RestaurantAffiliation,
RestaurantLocation, UserID)
VALUES
('Restaurant1', 'Location1', 6),
('Restaurant2', 'Location2', 7),
('Restaurant3', 'Location3', 8),
('Restaurant4', 'Location4', 9),
('Restaurant5', 'Location5', 10),
('Restaurant1', 'Location1', 11),
('Restaurant2', 'Location2', 12),
('Restaurant3', 'Location3', 13),
('Restaurant4', 'Location4', 14),
('Restaurant5', 'Location5', 15),
('Restaurant1', 'Location1', 16),
('Restaurant2', 'Location2', 17),
('Restaurant3', 'Location3', 18),
('Restaurant4', 'Location4', 19),
('Restaurant5', 'Location5', 20)";

$insertQuery9 = "INSERT INTO Leaderboard (LeaderboardCategory, Prize)
VALUES
('Global', 'Exclusive Cooking Set'),
('National', 'Cookbook Collection'),
('Regional', 'Kitchen Appliance Bundle')";

$insertQuery10 = "INSERT INTO EventLocation (Category, EntryFee, Location)
VALUES
('Cooking Competition', 50, 'City A'),
```

University of British Columbia, Vancouver

Department of Computer Science

```
('Baking Workshop', 30, 'City B'),
('Food Festival', 20, 'City C'),
('Culinary Tour', 40, 'City D'),
('Wine Tasting', 60, 'City E'),
('Sushi Making Class', 40, 'City F'),
('Pizza Workshop', 25, 'City G'),
('BBQ Cookout', 35, 'City H'),
('Dessert Masterclass', 45, 'City I'),
('Vegetarian Cooking Class', 30, 'City J'),
('Wine and Cheese Pairing', 55, 'City K'),
('Gourmet Dinner', 70, 'City L'),
('Street Food Festival', 15, 'City M'),
('Cocktail Mixology Workshop', 50, 'City N'),
('Farm-to-Table Experience', 65, 'City O')";
```

```
$insertQuery11 = "INSERT INTO EventDetails (Category, EntryFee, EventID,
Date)
```

```
VALUES
```

```
('Cooking Competition', 50, 'event1', '2024-04-15'),
('Baking Workshop', 30, 'event2', '2024-05-10'),
('Food Festival', 20, 'event3', '2024-06-20'),
('Culinary Tour', 40, 'event4', '2024-07-05'),
('Wine Tasting', 60, 'event5', '2024-08-12'),
('Sushi Making Class', 40, 'event6', '2024-09-05'),
('Pizza Workshop', 25, 'event7', '2024-10-10'),
('BBQ Cookout', 35, 'event8', '2024-11-20'),
('Dessert Masterclass', 45, 'event9', '2024-12-15'),
('Vegetarian Cooking Class', 30, 'event10', '2025-01-05'),
('Wine and Cheese Pairing', 55, 'event11', '2025-02-10'),
('Gourmet Dinner', 70, 'event12', '2025-03-20'),
('Street Food Festival', 15, 'event13', '2025-04-15'),
('Cocktail Mixology Workshop', 50, 'event14', '2025-05-10'),
('Farm-to-Table Experience', 65, 'event15', '2025-06-20')";
```

```
$insertQuery12 = "INSERT INTO ReviewDetails (TimePosted, Comment, Rating)
VALUES
```

```
('2023-01-01', 'Bad food!', 1),
('2023-02-15', 'Ok service!', 2),
('2023-03-20', 'Delicious dishes.', 3),
('2023-04-10', 'Nice ambiance.', 4),
('2023-05-05', 'Friendly staff.', 5)";
```

```
$insertQuery13 = "INSERT INTO Review (TimePosted, Comment, ReviewID)
VALUES
```

University of British Columbia, Vancouver

Department of Computer Science

```
('2023-01-01', 'Bad food!', 1),
('2023-02-15', 'Ok service!', 2),
('2023-03-20', 'Delicious dishes.', 3),
('2023-04-10', 'Nice ambiance.', 4),
('2023-05-05', 'Friendly staff.', 5)";

$insertQuery14 = "INSERT INTO CookingEquipment (Price, Category, Quality,
Brand)
VALUES
(50, 'Knives', 'High', 'Brand1'),
(100, 'Pots', 'Medium', 'Brand2'),
(80, 'Blenders', 'High', 'Brand3'),
(120, 'Ovens', 'High', 'Brand4'),
(60, 'Mixers', 'Medium', 'Brand5')";

$insertQuery15 = "INSERT INTO CookingEquipmentName (Name, Price, Category,
Quality)
VALUES
('Knife Set', 50, 'Knives', 'High'),
('Cookware Set', 100, 'Pots', 'Medium'),
('Blender', 80, 'Blenders', 'High'),
('Oven', 120, 'Ovens', 'High'),
('Mixer', 60, 'Mixers', 'Medium')";

$insertQuery16 = "INSERT INTO RecipeDetails (PublishDate, Title,
Description, Culture, Difficulty, Serving)
VALUES
('2024-01-01', 'Spaghetti Carbonara', 'Delicious dish with pasta, eggs,
cheese, and pancetta.', 'Italian', 'Intermediate', 4),
('2024-02-15', 'Guacamole', 'Simple and quick recipe for avocado lovers.',
'Mexican', 'Beginner', 2),
('2024-03-20', 'Sushi Rolls', 'Healthy and flavorful meal with fresh
seafood and vegetables.', 'Japanese', 'Advanced', 6),
('2024-04-10', 'Coq au Vin', 'Classic French recipe with chicken cooked in
wine.', 'French', 'Intermediate', 4),
('2024-05-05', 'Gulab Jamun', 'Traditional Indian dessert made with milk
solids and sugar syrup.', 'Indian', 'Expert', 8),
('2024-06-20', 'Kung Pao Chicken', 'Savory and spicy Chinese dish with
peanuts and vegetables.', 'Chinese', 'Intermediate', 5),
('2024-07-15', 'Greek Salad', 'Light and refreshing salad with tomatoes,
cucumbers, and feta cheese.', 'Mediterranean', 'Beginner', 3),
('2024-08-10', 'Clam Chowder', 'Hearty soup made with clams, potatoes, and
cream.', 'American', 'Intermediate', 6),
```

University of British Columbia, Vancouver

Department of Computer Science

```
('2024-09-05', 'Thai Green Curry', 'Spicy and flavorful Thai curry with chicken or vegetables.', 'Thai', 'Advanced', 7),
('2024-10-01', 'Black Forest Cake', 'Decadent chocolate cake with layers of cherries and whipped cream.', 'Swiss', 'Expert', 10),
('2024-11-10', 'Paneer Tikka Masala', 'Vegetarian Indian dish with marinated paneer in a creamy tomato sauce.', 'Indian', 'Intermediate', 4),
('2024-12-05', 'Lasagna', 'Classic Italian comfort food with layers of pasta, cheese, and meat sauce.', 'Italian', 'Beginner', 3),
('2025-01-20', 'Jerk Shrimp', 'Exotic Caribbean dish with spicy jerk seasoning.', 'Caribbean', 'Advanced', 6),
('2025-02-15', 'Agua Fresca', 'Refreshing Mexican drink made with fruit and water.', 'Mexican', 'Beginner', 2),
('2025-03-10', 'Sushi Burger', 'Innovative fusion recipe combining sushi and burger elements.', 'Asian Fusion', 'Intermediate', 5),
('2025-04-05', 'Alfredo Pasta', 'Rich and creamy pasta dish with Parmesan cheese sauce.', 'Italian', 'Expert', 8),
('2025-05-01', 'Falafel Wrap', 'Light and healthy Mediterranean wrap with falafel and veggies.', 'Mediterranean', 'Beginner', 2),
('2025-06-20', 'BBQ Ribs', 'Grilled American dish with smoky barbecue flavors.', 'American', 'Intermediate', 4),
('2025-07-15', 'Chili con Carne', 'Zesty Mexican dish with ground beef, beans, and spices.', 'Mexican', 'Advanced', 7),
('2025-08-10', 'Crème Brûlée', 'Decadent French dessert with a caramelized sugar topping.', 'French', 'Expert', 9)";
```

```
$insertQuery17 = "INSERT INTO Recipe (RecipeID, PublishDate, EstimatedTime, Title)
```

```
VALUES
```

```
(1, '2024-01-01', 60, 'Spaghetti Carbonara'),
(2, '2024-02-15', 30, 'Guacamole'),
(3, '2024-03-20', 90, 'Sushi Rolls'),
(4, '2024-04-10', 45, 'Coq au Vin'),
(5, '2024-05-05', 120, 'Gulab Jamun'),
(6, '2024-06-20', 75, 'Kung Pao Chicken'),
(7, '2024-07-15', 20, 'Greek Salad'),
(8, '2024-08-10', 60, 'Clam Chowder'),
(9, '2024-09-05', 90, 'Thai Green Curry'),
(10, '2024-10-01', 180, 'Black Forest Cake'),
(11, '2024-11-10', 40, 'Paneer Tikka Masala'),
(12, '2024-12-05', 30, 'Lasagna'),
(13, '2025-01-20', 120, 'Jerk Shrimp'),
(14, '2025-02-15', 15, 'Agua Fresca'),
(15, '2025-03-10', 60, 'Sushi Burger'),
(16, '2025-04-05', 90, 'Alfredo Pasta'),
```

University of British Columbia, Vancouver

Department of Computer Science

```
(17, '2025-05-01', 25, 'Falafel Wrap'),
(18, '2025-06-20', 45, 'BBQ Ribs'),
(19, '2025-07-15', 75, 'Chili con Carne'),
(20, '2025-08-10', 150, 'Crème Brûlée');"

$insertQuery18 = "INSERT INTO Video (Name, UploadTime, RecipeID, VideoURL,
Duration, Views)
VALUES
('Video1', '2024-01-01', 1, 'https://example.com/video1', 120, '1000'),
('Video2', '2024-02-15', 2, 'https://example.com/video1', 90, '800'),
('Video3', '2024-03-20', 3, 'https://example.com/video1', 150, '1200'),
('Video4', '2024-04-10', 4, 'https://example.com/video1', 60, '500'),
('Video5', '2024-05-05', 5, 'https://example.com/video1', 180, '1500');"

$insertQuery19 = "INSERT INTO Ingredient (Name, AllergenInfo)
VALUES
('Flour', 'Gluten'),
('Sugar', 'None'),
('Eggs', 'None'),
('Butter', 'Dairy'),
('Salt', 'None');"

$insertQuery20 = "INSERT INTO Follows (UserID1, UserID2)
VALUES
(1, 2),
(2, 3),
(3, 4),
(4, 5),
(5, 1),
(6, 7),
(7, 8),
(8, 9),
(9, 10),
(10, 6);"

$insertQuery21 = "INSERT INTO Participates (UserID, EventID)
VALUES
(1, 'event1'),
(1, 'event2'),
(1, 'event3'),
(1, 'event4'),
(1, 'event5'),
(1, 'event6'),
(1, 'event7'),
```

University of British Columbia, Vancouver

Department of Computer Science

```
(1, 'event8'),
(1, 'event9'),
(1, 'event10'),
(1, 'event11'),
(1, 'event12'),
(1, 'event13'),
(1, 'event14'),
(1, 'event15'),
(2, 'event1'),
(2, 'event2'),
(2, 'event3'),
(2, 'event4'),
(2, 'event5'),
(2, 'event6'),
(2, 'event7'),
(2, 'event8'),
(2, 'event9'),
(2, 'event10'),
(2, 'event11'),
(2, 'event12'),
(2, 'event13'),
(2, 'event14'),
(2, 'event15'),
(3, 'event1'),
(3, 'event2'),
(3, 'event3'),
(3, 'event4'),
(3, 'event5'),
(3, 'event6'),
(3, 'event7'),
(3, 'event8'),
(3, 'event9'),
(3, 'event10'),
(3, 'event11'),
(3, 'event12'),
(3, 'event13'),
(3, 'event14'),
(3, 'event15'),
(14, 'event2'),
(13, 'event3'),
(4, 'event4'),
(5, 'event5');" ;
```

University of British Columbia, Vancouver

Department of Computer Science

```
$insertQuery23 = "INSERT INTO UserRanking (UserID, LeaderboardCategory,
Points,Position)
VALUES
(1, 'Global', 100, 1),
(2, 'Global', 90,2),
(3, 'Global', 80,3),
(4, 'Global', 70,4),
(5, 'Global', 60,5),
(6, 'National', 100, 1),
(7, 'National', 90,2),
(8, 'National', 80,3),
(9, 'National', 70,4),
(10, 'National', 60,5),
(11, 'Regional', 100, 1),
(12, 'Regional', 90,2),
(13, 'Regional', 80,3),
(14, 'Regional', 70,4),
(15, 'Regional', 60,5)";
```

```
$insertQuery24 = "INSERT INTO Utilizes (Name, Price, Category, Quality,
RecipeID)
VALUES
('Knife Set', 50, 'Knives', 'High', 1),
('Cookware Set', 100, 'Pots', 'Medium', 2),
('Blender', 80, 'Blenders', 'High', 3),
('Oven', 120, 'Ovens', 'High', 4),
('Mixer', 60, 'Mixers', 'Medium', 5)";
```

```
$insertQuery25 = "INSERT INTO Posts (RecipeID, UserID)
VALUES
(1, 1),
(2, 2),
(3, 3),
(4, 4),
(5, 5)";
```

```
$connection->beginTransaction();
```

```
$connection->exec($insertQuery);
```

```
echo "<p>Inserted into UserDetails table.</p>\n";
```

```
$connection->exec($insertQuery2);
```

```
echo "<p>Inserted into AppUser table.</p>\n";
```


University of British Columbia, Vancouver

Department of Computer Science

```
$connection->exec($insertQuery5);  
echo "<p>Inserted into HomeCookDetails table.</p>\n";  
  
$connection->exec($insertQuery7);  
echo "<p>Inserted into ProfessionalChefSkill table.</p>\n";  
  
$connection->exec($insertQuery6);  
echo "<p>Inserted into ProfessionalChef table.</p>\n";  
  
$connection->exec($insertQuery9);  
echo "<p>Inserted into Leaderboard table.</p>\n";  
  
$connection->exec($insertQuery10);  
echo "<p>Inserted into EventLocation table.</p>\n";  
  
$connection->exec($insertQuery11);  
echo "<p>Inserted into EventDetails table.</p>\n";  
  
$connection->exec($insertQuery12);  
echo "<p>Inserted into ReviewDetails table.</p>\n";  
  
$connection->exec($insertQuery13);  
echo "<p>Inserted into Review table.</p>\n";  
  
$connection->exec($insertQuery14);  
echo "<p>Inserted into CookingEquipment table.</p>\n";  
  
$connection->exec($insertQuery15);  
echo "<p>Inserted into CookingEquipmentName table.</p>\n";  
  
$connection->exec($insertQuery16);  
echo "<p>Inserted into RecipeDetails table.</p>\n";  
  
$connection->exec($insertQuery17);  
echo "<p>Inserted into Recipe table.</p>\n";  
  
$connection->exec($insertQuery18);  
echo "<p>Inserted into Video table.</p>\n";  
  
$connection->exec($insertQuery19);  
echo "<p>Inserted into Ingredient table.</p>\n";  
  
$connection->exec($insertQuery20);
```

University of British Columbia, Vancouver

Department of Computer Science

```
echo "<p>Inserted into Follows table.</p>\n";

$connection->exec($insertQuery21);
echo "<p>Inserted into Participates table.</p>\n";

$connection->exec($insertQuery23);
echo "<p>Inserted into UserRanking table.</p>\n";

$connection->exec($insertQuery24);
echo "<p>Inserted into Utilizes table.</p>\n";

$connection->exec($insertQuery25);
echo "<p>Inserted into Posts table.</p>\n";

$connection->commit();
echo "<p>All data inserted successfully.</p>\n";

}
```

Summary of Project:

The project is a community-centric cooking recipe platform, allowing users to access recipes from professional chefs, attend competitive events to earn points, and review recipes. Real-life applications include assisting home cooks in meal planning, chefs sharing signature recipes, and users with dietary needs finding suitable recipes. The database functionality includes analyzing user preferences and event popularity, while the application platform uses JavaScript, HTML, CSS for the front-end, PHP for the backend, and Oracle for the Database Management System (DBMS).

SCHEMAS

UserDetails (*Username*, Password, NumberOfFollowers, NumberOfFollowing, Age)

AppUser (**UserID**, Username, Password)

HomeCook (**UserID**, FavouriteCuisine, HobbyistLevel)

University of British Columbia, Vancouver

Department of Computer Science

ProfessionalChefSkill (RestaurantAffiliation , RestaurantLocation, Certification)

ProfessionalChef (UserID, RestaurantAffiliation, RestaurantLocation)

Leaderboard (LeaderboardCategory, Prize)

EventLocation (Category, EntryFee, Location)

EventDetails (EventID, Category, EntryFee, Date)

ReviewDetails (TimePosted, Comment , Rating)

Review (ReviewID, TimePosted, Comment)

CookingEquipment (Price, Category, Quality, Brand)

CookingEquipmentName (Name, Price, Category, Quality)

RecipeDetails (PublishDate: DATE, Title: VARCHAR(20), Description: VARCHAR(200), Culture: VARCHAR(20) NOT NULL, Difficulty: VARCHAR(20) NOT NULL, Serving: INTEGER, PRIMARY KEY (PublishDate, Title))

Recipe (RecipeID, PublishDate, EstimatedTime, Title)

Video (Name, UploadTime, RecipeID, VideoURL, Duration, Views)

Ingredient (Name, AllergenInfo)

Contains (RecipeID, Name)

Follows (UserID1, UserID2)

Participates (UserID, EventID)

UserRanking (UserID, LeaderboardCategory, Points, Position)

Utilizes (Name, Price, Category, Quality, RecipeID)

Posts (RecipeID, UserID)

Reviews (ReviewID, RecipeID, UserID)

University of British Columbia, Vancouver

Department of Computer Science

The schema differed from the one we submitted earlier however the changes we made were not huge. It was mainly renaming of the decomposed relations and in some cases where there were issues with the normalization (extra attribute), the relations changed accordingly. There were no cases where there were major changes affecting the wider schema in the project.

Some example changes are as follows:

- Cooking equipment 1 and cooking equipment 2 were renamed to CookingEquipment and CookingEquipmentName
- We added a RecipeDescription attribute to the Recipes
- We added a videoURL attribute to Video and update the normalization accordingly

```
cpsc304=# \d
```

List of relations

Schema	Name	Type	Owner
public	appuser	table	postgres
public	contains	table	postgres
public	cookingequipment	table	postgres
public	cookingequipmentname	table	postgres
public	eventdetails	table	postgres
public	eventlocation	table	postgres
public	follows	table	postgres
public	homecook	table	postgres
public	ingredient	table	postgres
public	leaderboard	table	postgres
public	participates	table	postgres

University of British Columbia, Vancouver

Department of Computer Science

```
public | posts                | table | postgres
public | professionalchef        | table | postgres
public | professionalchefskill    | table | postgres
public | recipe                   | table | postgres
public | recipedetails            | table | postgres
public | review                   | table | postgres
public | reviewdetails            | table | postgres
public | reviews                  | table | postgres
public | userdetails              | table | postgres
public | userranking              | table | postgres
public | utilizes                  | table | postgres
public | video                     | table | postgres
```

(23 rows)

```
cpssc304=# select * from appuser;
```

```
username | password | userid
```

```
-----+-----+-----
```

```
jacky    | 110      |      1
```

```
safiullah | 320      |      2
```

```
mohammad | 310      |      3
```

```
seva     | 221      |      4
```

University of British Columbia, Vancouver

Department of Computer Science

raymond	304	5
nafis	304	6
gregor	110	7
clune	340	8
feeley	213	9
holmes	310	10
pottinger	304	11
wolfman	313	12
jordon	213	13
geoffrey	121	14
bradley	320	15
cinda	221	16
carter	210	17
patrice	313	18
ratna	295	19
anton	999	20

(20 rows)

```
cpssc304=# select * from جفٹبمغو ;
```

```
ERROR: syntax error at or near "?????????"
```

```
LINE 1: select * from ?????????;
```

```
cpssc304=# select * from contains;
```

```
  recipeid | name
```

```
-----+-----
```

```
(0 rows)
```

```
cpssc304=# select * from cookingequipment;
```

```
price | category | quality | brand
```

```
-----+-----+-----+-----
```

```
   50 | Knives   | High    | Brand1
```

```
  100 | Pots     | Medium  | Brand2
```

```
   80 | Blenders | High    | Brand3
```

```
  120 | Ovens    | High    | Brand4
```

```
   60 | Mixers   | Medium  | Brand5
```

```
(5 rows)
```

```
cpssc304=# select * from cookingequipmentname;
```

```
name      | price | category | quality
```

```
-----+-----+-----+-----
```

```
Knife Set |    50 | Knives   | High
```

Cookware Set		100		Pots		Medium
Blender		80		Blenders		High
Oven		120		Ovens		High
Mixer		60		Mixers		Medium
(5 rows)						
cpsc304=# select * from eventdetails;						
category		entryfee		eventid		date
-----+-----+-----+-----						
Cooking Competition		50		event1		2024-04-15
Baking Workshop		30		event2		2024-05-10
Food Festival		20		event3		2024-06-20
Culinary Tour		40		event4		2024-07-05
Wine Tasting		60		event5		2024-08-12
Sushi Making Class		40		event6		2024-09-05
Pizza Workshop		25		event7		2024-10-10
BBQ Cookout		35		event8		2024-11-20
Dessert Masterclass		45		event9		2024-12-15
Vegetarian Cooking Class		30		event10		2025-01-05
Wine and Cheese Pairing		55		event11		2025-02-10
Gourmet Dinner		70		event12		2025-03-20

Street Food Festival		15		event13		2025-04-15
Cocktail Mixology Workshop		50		event14		2025-05-10
Farm-to-Table Experience		65		event15		2025-06-20
(15 rows)						
cpsc304=# select * from eventlocation;						
category		entryfee		location		
-----+-----+						
Cooking Competition		50		City A		
Baking Workshop		30		City B		
Food Festival		20		City C		
Culinary Tour		40		City D		
Wine Tasting		60		City E		
Sushi Making Class		40		City F		
Pizza Workshop		25		City G		
BBQ Cookout		35		City H		
Dessert Masterclass		45		City I		
Vegetarian Cooking Class		30		City J		
Wine and Cheese Pairing		55		City K		
Gourmet Dinner		70		City L		
Street Food Festival		15		City M		

University of British Columbia, Vancouver

Department of Computer Science

```
Cocktail Mixology Workshop |          50 | City N
```

```
Farm-to-Table Experience   |          65 | City O
```

(15 rows)

```
cpsc304=# select * from follows;
```

```
userid1 | userid2
```

```
-----+-----
```

```
1 | 2
```

```
2 | 3
```

```
3 | 4
```

```
4 | 5
```

```
5 | 1
```

```
6 | 7
```

```
7 | 8
```

```
8 | 9
```

```
9 | 10
```

```
10 | 6
```

(10 rows)

```
cpsc304=# select * from homecook;
```

University of British Columbia, Vancouver

Department of Computer Science

```
favouritecuisine | hobbyistlevel | userid
-----+-----+-----
Italian          | Amateur          |      1
Mexican          | Intermediate      |      2
Japanese         | Advanced          |      3
Italian          | Amateur          |      4
British          | Advanced          |      5
(5 rows)
```

```
cpssc304=# select * from ingredient;
```

```
name | allergeninfo
-----+-----
Flour | Gluten
Sugar | None
Eggs  | None
Butter | Dairy
Salt  | None
(5 rows)
```

```
cpssc304=# select * from leaderboard;
```

```
leaderboardcategory | prize
-----+-----
Global              | Exclusive Cooking Set
National            | Cookbook Collection
Regional            | Kitchen Appliance Bundle
(3 rows)

cpsc304=# \d

                List of relations

 Schema |          Name          | Type  | Owner
-----+-----+-----+-----
 public | appuser                 | table | postgres
 public | contains                | table | postgres
 public | cookingequipment        | table | postgres
 public | cookingequipmentname    | table | postgres
 public | eventdetails            | table | postgres
 public | eventlocation           | table | postgres
 public | follows                 | table | postgres
 public | homecook                | table | postgres
 public | ingredient               | table | postgres
 public | leaderboard              | table | postgres
```

University of British Columbia, Vancouver

Department of Computer Science

```
public | participates      | table | postgres
public | posts                  | table | postgres
public | professionalchef        | table | postgres
public | professionalchefskill    | table | postgres
public | recipe                   | table | postgres
public | recipedetails            | table | postgres
public | review                   | table | postgres
public | reviewdetails            | table | postgres
public | reviews                  | table | postgres
public | userdetails              | table | postgres
public | userranking              | table | postgres
public | utilizes                  | table | postgres
public | video                    | table | postgres
```

(23 rows)

```
cp304=# select * from participates;
```

```
userid | eventid
```

```
-----+-----
```

```
1 | event1
```

```
1 | event2
```

```
1 | event3
```

1 | event4

1 | event5

1 | event6

1 | event7

1 | event8

1 | event9

1 | event10

1 | event11

1 | event12

1 | event13

1 | event14

1 | event15

2 | event1

2 | event2

2 | event3

2 | event4

2 | event5

2 | event6

2 | event7

2 | event8

2 | event9

2 | event10

2 | event11

2 | event12

2 | event13

2 | event14

2 | event15

3 | event1

3 | event2

3 | event3

3 | event4

3 | event5

3 | event6

3 | event7

3 | event8

3 | event9

3 | event10

3 | event11

3 | event12

3 | event13

3 | event14

3 | event15

14 | event2

13 | event3

University of British Columbia, Vancouver

Department of Computer Science

4 | event4

5 | event5

(49 rows)

cpssc304=# \d

List of relations

Schema	Name	Type	Owner
public	appuser	table	postgres
public	contains	table	postgres
public	cookingequipment	table	postgres
public	cookingequipmentname	table	postgres
public	eventdetails	table	postgres
public	eventlocation	table	postgres
public	follows	table	postgres
public	homecook	table	postgres
public	ingredient	table	postgres
public	leaderboard	table	postgres
public	participates	table	postgres
public	posts	table	postgres
public	professionalchef	table	postgres

University of British Columbia, Vancouver

Department of Computer Science

```
public | professionalchefskill | table | postgres
public | recipe                | table | postgres
public | recipedetails            | table | postgres
public | review                    | table | postgres
public | reviewdetails              | table | postgres
public | reviews                    | table | postgres
public | userdetails                | table | postgres
public | userranking                 | table | postgres
public | utilizes                    | table | postgres
public | video                       | table | postgres
```

(23 rows)

```
cpssc304=# select * from posts;
```

```
recipeid | userid
```

```
-----+-----
```

```
1 | 1
```

```
2 | 2
```

```
3 | 3
```

```
4 | 4
```

```
5 | 5
```

(5 rows)

```
cpssc304=# select * from professionalchef;
```

```
  restaurantaffiliation | restaurantlocation | userid
```

```
-----+-----+-----
```

```
Restaurant1           | Location1           |      6
```

```
Restaurant2           | Location2           |      7
```

```
Restaurant3           | Location3           |      8
```

```
Restaurant4           | Location4           |      9
```

```
Restaurant5           | Location5           |     10
```

```
Restaurant1           | Location1           |     11
```

```
Restaurant2           | Location2           |     12
```

```
Restaurant3           | Location3           |     13
```

```
Restaurant4           | Location4           |     14
```

```
Restaurant5           | Location5           |     15
```

```
Restaurant1           | Location1           |     16
```

```
Restaurant2           | Location2           |     17
```

```
Restaurant3           | Location3           |     18
```

```
Restaurant4           | Location4           |     19
```

```
Restaurant5           | Location5           |     20
```

```
(15 rows)
```

```
cpssc304=# select * from professionalchefskill;

restaurantaffiliation | restaurantlocation | certification
-----+-----+-----
1 Restaurant1          | Location1          | Chef Certification
2 Restaurant2          | Location2          | Chef Certification
3 Restaurant3          | Location3          | Chef Certification
4 Restaurant4          | Location4          | Chef Certification
5 Restaurant5          | Location5          | Chef Certification
(5 rows)
```

```
cpssc304=# select * from recipe;

recipeid | publishdate | estimatedtime | title
-----+-----+-----+-----
1 | 2024-01-01 | 60 | Spaghetti Carbonara
2 | 2024-02-15 | 30 | Guacamole
3 | 2024-03-20 | 90 | Sushi Rolls
4 | 2024-04-10 | 45 | Coq au Vin
```

Department of Computer Science

(20 rows)

```
publishdate | title |
description | culture |
difficulty | serving
```

-----+-----+-----		
-----+-----		
+-----+-----		
2024-01-01	Spaghetti Carbonara	Delicious dish with pasta, eggs, cheese, and pancetta.
Intermediate	4	Italian
2024-02-15	Guacamole	Simple and quick recipe for avocado lovers.
Beginner	2	Mexican
2024-03-20	Sushi Rolls	Healthy and flavorful meal with fresh seafood and vegetables.
Advanced	6	Japanese
2024-04-10	Coq au Vin	Classic French recipe with chicken cooked in wine.
Intermediate	4	French
2024-05-05	Gulab Jamun	Traditional Indian dessert made with milk solids and sugar syrup.
Expert	8	Indian
2024-06-20	Kung Pao Chicken	Savory and spicy Chinese dish with peanuts and vegetables.
Intermediate	5	Chinese
2024-07-15	Greek Salad	Light and refreshing salad with tomatoes, cucumbers, and feta cheese.
Beginner	3	Mediterranean
2024-08-10	Clam Chowder	Hearty soup made with clams, potatoes, and cream.
Intermediate	6	American
2024-09-05	Thai Green Curry	Spicy and flavorful Thai curry with chicken or vegetables.
Advanced	7	Thai
2024-10-01	Black Forest Cake	Decadent chocolate cake with layers of cherries and whipped cream.
Expert	10	Swiss

Department of Computer Science

(20 rows)

University of British Columbia, Vancouver

Department of Computer Science

```
cpssc304=# select * from review;
```

timeposted	comment	reviewid
2023-01-01	Bad food!	1
2023-02-15	Ok service!	2
2023-03-20	Delicious dishes.	3
2023-04-10	Nice ambiance.	4
2023-05-05	Friendly staff.	5

(5 rows)

```
cpssc304=# select * from reviewdetails;
```

timeposted	comment	rating
2023-01-01	Bad food!	1
2023-02-15	Ok service!	2
2023-03-20	Delicious dishes.	3
2023-04-10	Nice ambiance.	4
2023-05-05	Friendly staff.	5

(5 rows)

University of British Columbia, Vancouver

Department of Computer Science

```
cpssc304=# select * from userdetails;
```

```
  numberoffollowers | numberoffollowing | age | username |  
password
```

```
-----+-----+-----+-----+-----  
----
```

100		50		25		jacky		110
200		75		30		safiullah		320
150		60		28		mohammad		310
120		45		22		seva		221
80		35		27		raymond		304
145		90		89		nafis		304
89		453		43		gregor		110
57		329		71		clune		340
67		78		56		feeley		213
98		46		32		holmes		310
87		20		102		pottinger		304
68		198		125		wolfman		313
56		980		98		jordon		213
110		589		75		geoffrey		121
120		437		32		bradley		320
89		84938		43		cinda		221
96		473827483		87		carter		210

University of British Columbia, Vancouver

Department of Computer Science

48		2490		12		patrice		313
----	--	------	--	----	--	---------	--	-----

82		437843		35		ratna		295
----	--	--------	--	----	--	-------	--	-----

102		3848		23		anton		999
-----	--	------	--	----	--	-------	--	-----

(20 rows)

```
cpssc304=# select * from userranking;
```

userid		leaderboardcategory		points		position
--------	--	---------------------	--	--------	--	----------

-----+-----+-----+-----

1		Global		100		1
---	--	--------	--	-----	--	---

2		Global		90		2
---	--	--------	--	----	--	---

3		Global		80		3
---	--	--------	--	----	--	---

4		Global		70		4
---	--	--------	--	----	--	---

5		Global		60		5
---	--	--------	--	----	--	---

6		National		100		1
---	--	----------	--	-----	--	---

7		National		90		2
---	--	----------	--	----	--	---

8		National		80		3
---	--	----------	--	----	--	---

9		National		70		4
---	--	----------	--	----	--	---

10		National		60		5
----	--	----------	--	----	--	---

11		Regional		100		1
----	--	----------	--	-----	--	---

12		Regional		90		2
----	--	----------	--	----	--	---

13		Regional		80		3
----	--	----------	--	----	--	---

University of British Columbia, Vancouver

Department of Computer Science

14		Regional		70		4
----	--	----------	--	----	--	---

15		Regional		60		5
----	--	----------	--	----	--	---

(15 rows)

```
cpssc304=# select * from utilizes;
```

name		price		category		quality		recipeid
------	--	-------	--	----------	--	---------	--	----------

-----+-----+-----+-----+-----

Knife Set		50		Knives		High		1
-----------	--	----	--	--------	--	------	--	---

Cookware Set		100		Pots		Medium		2
--------------	--	-----	--	------	--	--------	--	---

Blender		80		Blenders		High		3
---------	--	----	--	----------	--	------	--	---

Oven		120		Ovens		High		4
------	--	-----	--	-------	--	------	--	---

Mixer		60		Mixers		Medium		5
-------	--	----	--	--------	--	--------	--	---

(5 rows)

```
cpssc304=# select * from video;
```

name		uploadtime		recipeid		videourl	
duration		views					

-----+-----+-----+-----+-----

-----+-----

Video1		2024-01-01		1		https://example.com/video1	
120		1000					

University of British Columbia, Vancouver

Department of Computer Science

```
Video2 | 2024-02-15 |          2 | https://example.com/video1 |
90 | 800

Video3 | 2024-03-20 |          3 | https://example.com/video1 |
150 | 1200

Video4 | 2024-04-10 |          4 | https://example.com/video1 |
60 | 500

Video5 | 2024-05-05 |          5 | https://example.com/video1 |
180 | 1500

(5 rows)
```

Queries (and their location)

INSERT:

From createdRecipeForm.php:

Line 22:

```
$insertQueryRecipeDetails = "INSERT INTO RecipeDetails (PublishDate,
Title, Description, Culture, Difficulty, Serving)
VALUES
(:publishedDate, :recipeTitle, :recipeDescription, :culture,
:difficultyLevel, :servings)";
```

Line 47:

```
$insertQueryRecipe = "INSERT INTO Recipe (RecipeID, PublishDate,
EstimatedTime, Title)
VALUES
(:recipeID, :publishedDate, :estimatedTime, :recipeTitle)";
```

Line 65:

University of British Columbia, Vancouver

Department of Computer Science

```
$insertPosts = "INSERT INTO Posts (RecipeID, UserID)
VALUES
(:recipeID, :userID)";
```

Line 83:

```
$insertStmt = $connection->prepare("INSERT INTO Utilizes (Name, Price,
Category, Quality, RecipeID) VALUES (:name, :price, :category, :quality,
:recipeID)");
```

Line 97:

```
$insertStmt = $connection->prepare("INSERT INTO Utilizes (Name, Price,
Category, Quality, RecipeID) VALUES (:name, :price, :category, :quality,
:recipeID)");
```

Line 114:

```
$insertStmt = $connection->prepare("INSERT INTO Contains (RecipeID,Name)
VALUES (:recipeID,:name)");
```

Line 123:

```
$insertQuery18 = $connection->prepare("INSERT INTO Video (Name,
UploadTime, RecipeID, VideoURL, Duration, Views) VALUES (:name,
:uploadTime, :recipeId, :videoURL, :duration, 0)");
```

From registerhomecook.php

Line 23:

```
$sql = "INSERT INTO UserDetails (NumberOfFollowers, NumberOfFollowing,
Age, Username, Password) VALUES (0,0,:age,:username, :password)";
```

Line 33:

```
$sql = "INSERT INTO AppUser (Username, Password, UserID) VALUES
(:username, :password, :UserID)";
```

Line 41:

```
$sql = "INSERT INTO HomeCook (FavouriteCuisine, HobbyistLevel, UserID)
VALUES (:favouriteCuisine, 'Amateur', :UserID)";
```

From registerprofessionalchef.php:

Line 36:

```
$sql = "INSERT INTO UserDetails (NumberOfFollowers, NumberOfFollowing,
Age, Username, Password) VALUES (0,0,:age,:username, :password)";
```

Line 46:

University of British Columbia, Vancouver

Department of Computer Science

```
$sql = "INSERT INTO AppUser (Username, Password, UserID) VALUES  
(:username, :password, :UserID)";
```

Line 56:

```
$sql = "INSERT INTO ProfessionalChefSkill (RestaurantAffiliation,  
RestaurantLocation, Certification) VALUES  
(:RestaurantAffiliation, :RestaurantLocation, :Certification)";
```

Line 63:

```
$sql = "INSERT INTO ProfessionalChef (RestaurantAffiliation,  
RestaurantLocation, UserID) VALUES  
(:RestaurantAffiliation, :RestaurantLocation, :UserID)";
```

INSERT

From deleteuser.php

Line 56:

```
$delete = $connection->prepare("DELETE FROM userdetails  
WHERE username = :username  
AND password = :password");
```

UPDATE

From updateRecipe.php:

Line 16:

```
$updateRecipeDetails = $connection->prepare("  
UPDATE RecipeDetails  
SET Description = :description,  
    Culture = :culture,  
    Difficulty = :difficulty,  
    Serving = :serving  
FROM Recipe  
WHERE RecipeDetails.PublishDate = Recipe.PublishDate  
AND RecipeDetails.Title = Recipe.Title  
AND Recipe.RecipeID = :recipeId
```

Line 37:

University of British Columbia, Vancouver

Department of Computer Science

```
$updateRecipe = $connection->prepare("UPDATE Recipe SET EstimatedTime = :estimatedTime WHERE RecipeID = :recipeId");
```

SELECT

From recipefilter.php

Line 114:

```
$sqlConditionStr = implode(' ', $sqlConditions);  
    $sql = "SELECT * FROM RecipeDetails r JOIN Recipe r2 ON  
r2.title = r.title AND r2.publishdate = r.publishdate WHERE  
$sqlConditionStr";
```

PROJECTION

From recipefilter.php

Line 77:

```
$selectedColumnsString = implode(', ', $selectedColumns);  
  
    $sql = "SELECT $selectedColumnsString FROM RecipeDetails r JOIN  
Recipe r2 ON r2.title = r.title AND r2.publishdate = r.publishdate";
```

JOIN

From recipefilter.php

Line 15:

```
if ($culture === 'any') {  
    $sql = "SELECT * FROM recipedetails r  
        JOIN recipe r2 ON r2.title = r.title AND r2.publishdate =  
r.publishdate  
        WHERE r.serving >= :serving";
```

Line 22:

```
$sql = "SELECT *
        FROM recipedetails r, (
            SELECT culture
            FROM recipedetails
            GROUP BY culture
            HAVING culture = :culture AND MIN(serving) >=
:serving
        ) c, recipe r2
        WHERE c.culture = r.culture
        AND r2.title = r.title
        AND r2.publishdate = r.publishdate";
}
```

AGGREGATION WITH HAVING

From userfilter.php

Line 29:

```
$checkStmt2 = $connection->prepare("SELECT u.age, COUNT(*)
                                     FROM userdetails u
                                     GROUP BY u.age
                                     HAVING MIN(u.age) >= :age2");
```

NESTED AGGREGATION WITH GROUP BY

From userfilter.php

Line 16:

```
$checkStmt = $connection->prepare("SELECT u.age, COUNT(*)
                                     FROM userdetails u
                                     WHERE u.age >= :age
                                     GROUP BY u.age
                                     HAVING AVG(numberoffollowers) >= (SELECT
AVG(numberoffollowers)
```

University of British Columbia, Vancouver

Department of Computer Science

```
userdetails u2)");
```

FROM

Users Statistics

[Home](#)

Enter age amount:

[Search](#)

Find out how many people are in each age group where the average number of followers for users older than a specific age is higher than the average number of followers for all users in every age group. (Find which age group is more popular based on the average number of followers across all age groups) :

Find the number of users and their age that are equal to or above the user-inputted age:

Find the average number of followers and number of following user of an age group:

Users Statistics

[Home](#)

Enter age amount:

[Search](#)

Find out how many people are in each age group where the average number of followers for users older than a specific age is higher than the average number of followers for all users in every age group. (Find which age group is more popular based on the average number of followers across all age groups) :

Age: 89, Number of People: 1
Age: 75, Number of People: 1

Find the number of users and their age that are equal to or above the user-inputted age:

Age: 87, Number of People: 1
Age: 71, Number of People: 1
Age: 98, Number of People: 1
Age: 125, Number of People: 1
Age: 56, Number of People: 1
Age: 89, Number of People: 1
Age: 102, Number of People: 1
Age: 75, Number of People: 1

Find the average number of followers and number of following user of an age group:

Age: 87, Average Number of Follows: 473827483, Average Number of Followers: 96
Age: 71, Average Number of Follows: 329, Average Number of Followers: 57
Age: 125, Average Number of Follows: 198, Average Number of Followers: 68
Age: 89, Average Number of Follows: 90, Average Number of Followers: 145

DIVISION:

From eventsfilter.php

Line 11:

```
$checkStmt = $connection->prepare("SELECT *  
                                FROM userdetails u,  
                                (SELECT *  
                                FROM appuser a  
                                WHERE NOT EXISTS
```


University of British Columbia, Vancouver

Department of Computer Science

```
((SELECT e.eventid
FROM eventdetails e)
EXCEPT
(SELECT p.eventid
FROM participates p
WHERE p.userid = a.userid))) t
WHERE t.username = u.username
AND t.password = u.password");
```

HALL OF FAME

Users Who Have In Participated Every Single Event:

Name: Jacky, Number Of Following: 50, Number Of Followers: 100, Age: 25

Name: Safiullah, Number Of Following: 75, Number Of Followers: 200, Age: 30

Name: Mohammad, Number Of Following: 60, Number Of Followers: 150, Age: 28

[Home](#)