

# RecipeDelight Database Project

## ITCS 3160 - 001

Fall 2019

### Database Dudes

Isaac Altice

Jon Labrador

Laiken Cleary

Kyungmi Seung

December 4, 2019

# **Business Rules and Assumptions:**

1. Person – Chef/Student is a generalization-specialization relationship (superclass/subclass)
2. A chef may be simultaneously assigned to multiple recipe projects. A project has a chef designated as the Lead Chef. Egos being what they are, there may be multiple Lead Chefs
3. A student may be simultaneously assigned to multiple recipe projects.
4. A recipe is designated as available for download or for inclusion in an eCookBook when it is published as indicated by its publication date.
5. A chef or a student may be associated with multiple recipe projects.
6. A chef may be the only person assigned to a recipe project.
7. A student may not be the only person assigned to a recipe project.
8. Many recipes contain some of the same ingredient(s), and it is advisable for standardization that ensures references to an ingredient name are the same across all recipes.
9. An ingredient may be listed only once in a given recipe.
10. You may choose to express ingredient quantities in metric or English units.
11. Quantities should be expressed precisely. For instance, 1/2 stored as a float field might be rendered as 0.4999999 instead of 0.5. Therefore, quantity fields should probably not be the float data type.
12. Quantities should be usable in arithmetic computations. For instance, if the Number Served field of a recipe needs to be doubled, you may assume that all ingredient Quantities may be multiplied by two.
13. Awards are associated with recipes.
14. A recipe can win an award only once. Awards may be differentiated by time. A recipe could win both the Best Entre 2019 and Best Entre 2020 awards but could not win the Best Entre 2019 more than once.
15. A recipe can be included in a given eCookBook only once.
16. A recipe can be included in multiple eCookBooks.
17. Chef reports
  1. Listing unpublished recipe projects by Lead Chief.
  2. Listing published recipe projects by Lead Chief.
  3. Listing all recipe projects and include all assigned chefs.
18. Student reports
  1. Listing all students and their published recipe projects with assigned chefs.
19. Management report
  1. Listing all chefs including names and salaries.

2. Listing of all students with names, a count of assigned, unpublished recipe projects and expected graduation date.

#### 20. Recipes and eCookBooks

1. List each recipe with Recipe Name, Lead Chef, and number of times it has been downloaded.

2. List each Recipe Name with its Ingredients. Include the Quantity and Units.

3. List each Recipe Name with its Instructions.

4. List each eCookBook with eCookBook Name and number of recipes in it.

#### 21. Awards

1. List all recipes that have received awards including the name of the award. Do this report two ways. Once with a natural join and once with the JOIN statement.

## Team Members and Project Assignments

Isaac Altice: EERD, Relational Diagram, Update EERD and Relational Diagram, Data Dictionary

Jon Ledbetter: Rules and Assumptions, SQL Queries and Updates, Advance SQL features and descriptions

Laiken Cleary: Project formatting and assignment, Sample query Output, database testing

Kyungmi Seung: EERD, Sample Test Data, Final project aesthetics, Executive Summary

## Drop & Create Queries :

```
DROP DATABASE IF EXISTS recipe_delight;
CREATE DATABASE IF NOT EXISTS recipe_delight;
USE recipe_delight;

DROP TABLE IF EXISTS Person;
CREATE TABLE Person (
    PersonID INTEGER unsigned NOT NULL auto_increment,
    Email VARCHAR (64) default NULL,
    FirstName VARCHAR (64) default NULL,
    LastName VARCHAR (64) default NULL,
    StreetAddress VARCHAR (64) default NULL,
    City VARCHAR (64) default NULL,
    State VARCHAR (2) default NULL,
    Zipcode INTEGER default NULL,
    PRIMARY KEY (PersonID)
) AUTO_INCREMENT=1;

DROP TABLE IF EXISTS Chef;
CREATE TABLE Chef(
    ChefID INTEGER unsigned NOT NULL auto_increment,
    Specialty VARCHAR (64) default NULL,
    Salary INTEGER default NULL,
    FKPersonID INTEGER unsigned NOT NULL,
    FOREIGN KEY (FKPersonID) REFERENCES Person (personID),
    PRIMARY KEY (ChefID)
) AUTO_INCREMENT=1;

DROP TABLE IF EXISTS Student;
CREATE TABLE Student(
    StudentID INTEGER unsigned NOT NULL auto_increment,
    Major VARCHAR (64) default NULL,
    ExpectedGraduationDate Date default NULL,
    FKPersonID INTEGER unsigned NOT NULL,
    FOREIGN KEY (FKPersonID) REFERENCES person (personID),
    PRIMARY KEY (StudentID)
) AUTO_INCREMENT=1;
```



```
DROP TABLE IF EXISTS LeadChef;
CREATE TABLE LeadChef(
LeadChefID INTEGER unsigned NOT NULL auto_increment,
FKLeadChefID INTEGER unsigned NOT NULL,
PRIMARY KEY (LeadChefID),
foreign key (FKLeadChefID) references Chef(ChefID)
)AUTO_INCREMENT =1;
```

```
DROP TABLE IF EXISTS DevelopmentProject;
CREATE TABLE DevelopmentProject(
ProjectID INTEGER unsigned NOT NULL auto_increment,
Name VARCHAR(64),
PRIMARY KEY (ProjectID)
) AUTO_INCREMENT=1;
```

```
DROP TABLE IF EXISTS Student_DevelopmentProject;
CREATE TABLE Student_DevelopmentProject(
StudentID INTEGER unsigned NOT NULL,
ProjectID INTEGER unsigned NOT NULL,
PRIMARY KEY (StudentID,ProjectID),
FOREIGN KEY (StudentID) REFERENCES Student(StudentID),
FOREIGN KEY (ProjectID) REFERENCES DevelopmentProject(ProjectID)
);
```

```
DROP TABLE IF EXISTS LeadChef_DevelopmentProject;
CREATE TABLE LeadChef_DevelopmentProject(
LeadChefID INTEGER unsigned NOT NULL,
ProjectID INTEGER unsigned NOT NULL,
PRIMARY KEY (LeadChefID,ProjectID),
FOREIGN KEY (LeadChefID) REFERENCES LeadChef(LeadChefID),
FOREIGN KEY (ProjectID) REFERENCES DevelopmentProject(ProjectID)
);
```

```
DROP TABLE IF EXISTS Chef_DevelopmentProject;
CREATE TABLE Chef_DevelopmentProject(
ChefID INTEGER unsigned NOT NULL,
ProjectID INTEGER unsigned NOT NULL,
PRIMARY KEY (ChefID,ProjectID),
FOREIGN KEY (ChefID) REFERENCES Chef(ChefID),
FOREIGN KEY (ProjectID) REFERENCES DevelopmentProject(ProjectID)
);
```



```

DROP TABLE IF EXISTS Recipe;
CREATE TABLE Recipe(
    RecipeID INTEGER unsigned NOT NULL auto_increment,
    Name VARCHAR (64) default NULL,
    Downloads INTEGER default NULL,
    PublicationDate Date default NULL,
    Category VARCHAR (64) default NULL,
    Cuisine VARCHAR (64) default NULL,
    NumberServed INTEGER default NULL,
    PRIMARY KEY (RecipeID)
) AUTO_INCREMENT=1;

DROP TABLE IF EXISTS eCookBook;
CREATE TABLE eCookBook(
    eCookBookID INTEGER unsigned NOT NULL AUTO_INCREMENT,
    NumberOfRecipes INTEGER default NULL,
    Theme VARCHAR (64) default NULL,
    Name VARCHAR (64) default NULL,
    PRIMARY KEY (eCookBookID)
) AUTO_INCREMENT=1;

DROP TABLE IF EXISTS SpecialtyEquipment;
CREATE TABLE SpecialtyEquipment(
    EquipmentID INTEGER unsigned NOT NULL auto_increment,
    Name VARCHAR (64) default NULL,
    PRIMARY KEY(EquipmentID)
) AUTO_INCREMENT=1;

DROP TABLE IF EXISTS Ingredient;
CREATE TABLE Ingredient(
    IngredientID INTEGER unsigned NOT NULL auto_increment,
    Units VARCHAR (64) default NULL,
    Name VARCHAR (64) default NULL,
    PRIMARY KEY(IngredientID)
) AUTO_INCREMENT = 1;

DROP TABLE IF EXISTS Instruction;
CREATE TABLE Instruction(
    InstructionStepID INTEGER unsigned NOT NULL auto_increment,
    Instruction VARCHAR (255) default NULL,
    PRIMARY KEY(InstructionStepID)
) AUTO_INCREMENT = 1;

```



```

DROP TABLE IF EXISTS Award;
CREATE TABLE Award(
AwardID INTEGER unsigned NOT NULL auto_increment,
Name VARCHAR (64) default NULL,
PRIMARY KEY (AwardID)
) AUTO_INCREMENT =1;
DROP TABLE IF EXISTS Recipe_LeadChef;
CREATE TABLE Recipe_leadChef(
RecipeID INTEGER unsigned NOT NULL,
LeadChefID INTEGER unsigned NOT NULL,
PRIMARY KEY (RecipeID,LeadChefID),
FOREIGN KEY (RecipeID) REFERENCES Recipe(RecipeID),
FOREIGN KEY (LeadChefID) REFERENCES LeadChef(LeadChefID)
);

DROP TABLE IF EXISTS Recipe_Chef;
CREATE TABLE Recipe_Chef(
RecipeID INTEGER unsigned NOT NULL,
ChefID INTEGER unsigned NOT NULL,
PRIMARY KEY (RecipeID, ChefID),
FOREIGN KEY (RecipeID) REFERENCES Recipe(RecipeID),
FOREIGN KEY (ChefID) REFERENCES Chef(ChefID)
);

DROP TABLE IF EXISTS Recipe_Student;
CREATE TABLE Recipe_Student(
RecipeID INTEGER unsigned NOT NULL,
StudentID INTEGER unsigned NOT NULL,
PRIMARY KEY (RecipeID, StudentID),
FOREIGN KEY (RecipeID) REFERENCES Recipe(RecipeID),
FOREIGN KEY (StudentID) REFERENCES Student(StudentID)
);

DROP TABLE IF EXISTS Recipe_Award;
CREATE TABLE Recipe_Award(
RecipeID INTEGER unsigned NOT NULL DEFAULT 0,
AwardID INTEGER unsigned NOT NULL DEFAULT 0,
Date Date NOT NULL,
PRIMARY KEY (RecipeID, AwardID, Date),
FOREIGN KEY (RecipeID) REFERENCES Recipe(RecipeID),
FOREIGN KEY (AwardID) REFERENCES Award(AwardID)
);

```



```
DROP TABLE IF EXISTS Recipe_SpecialtyEquipment;
CREATE TABLE Recipe_SpecialtyEquipment(
RecipeID INTEGER unsigned NOT NULL,
EquipmentID INTEGER unsigned NOT NULL,
PRIMARY KEY (RecipeID, EquipmentID),
FOREIGN KEY (RecipeID) REFERENCES Recipe(RecipeID),
FOREIGN KEY (EquipmentID) REFERENCES SpecialtyEquipment(EquipmentID)
);

DROP TABLE IF EXISTS Recipe_Ingredient;
CREATE TABLE Recipe_Ingredient(
RecipeID INTEGER unsigned NOT NULL,
IngredientID INTEGER unsigned NOT NULL,
Quantity INTEGER default NULL,
PRIMARY KEY (RecipeID, IngredientID),
FOREIGN KEY (RecipeID) REFERENCES Recipe(RecipeID),
FOREIGN KEY (IngredientID) REFERENCES Ingredient(IngredientID)
);

DROP TABLE IF EXISTS Recipe_Instruction;
CREATE TABLE Recipe_Instruction(
RecipeID INTEGER unsigned NOT NULL,
InstructionStepID INTEGER unsigned NOT NULL,
PRIMARY KEY (RecipeID, InstructionStepID),
FOREIGN KEY (RecipeID) REFERENCES Recipe(RecipeID),
FOREIGN KEY (InstructionStepID) REFERENCES Instruction(InstructionStepID)
);

DROP TABLE IF EXISTS Recipe_eCookBook;
CREATE TABLE Recipe_eCookBook(
RecipeID INTEGER unsigned NOT NULL,
eCookBookID INTEGER unsigned NOT NULL,
PRIMARY KEY (RecipeID, eCookBookID),
FOREIGN KEY (RecipeID) REFERENCES Recipe(RecipeID),
FOREIGN KEY (eCookBookID) REFERENCES eCookBook(eCookBookID)
);
```



# Insert Data Queries (Test Data) :

```
INSERT INTO person
(PersonID,Email,FirstName,LastName,StreetAddress,City,State,Zipcode)
VALUES
(1,"egestas.Duis.ac@rutmlorem.org","Connor", "English","598-2101
Aliquet, Road","Beaumont","NC","09367"),
(2,"nec@interdumNunc.co.uk","Alvin", "Brennan","P.O. Box 533, 235 Donec
Rd.","Oostkamp","VA","48063"),
(3,"interdum.enim.non@sit.edu","Reece", "Tyler","Ap #591-8904 Ut
Street","Burnaby","TN","468079"),
(4,"lorem.ut.aliquam@cursuspurusNullam.com","Aquila", "Kerr","P.O. Box
852, 7602 Aliquet, Rd.","Neuville","SC","85612"),
(5,"tellus.imperdiet@neque.org","Troy", "Reeves","700-4472 Posuere
Rd.","Attigliano","GA","52343"),
(6,"non.lacinia.at@placerataugueSed.co.uk","Merritt", "Grant","Ap
#498-4861 Urna. St.","Lang","AL","32137"),
(7,"arcu@Donectempuslorem.com","Andrew", "Massey","P.O. Box 105, 778
Laoreet St.","Antalya","NC","82362"),
(8,"dui.Cras.pellentesque@Donecfelisorci.org","Vaughan", "Obrien","P.O.
Box 436, 7616 Enim, Av.", "Mörfelden-Walldorf","NC","77918"),
(9,"Aliquam.vulputate@Maecenasornare.ca","Alvin", "Little","Ap 884-6004
Non, Rd.","Maastricht","SC","238663"),
(10,"Mauris.eu@magnisdis.com","Nathan", "Kaufman","Ap 843-2756 Lectus
Street","Cabano","NC","50619"),
(11,"sem.ut.dolor@elitfermentumrisus.edu","Jackson", "Ashley","405-2804
Dolor. Rd.","Radebeul","NC","0899595"),
(12,"lacus@diamatpretium.ca","Reece", "Padilla","Ap 108-3435 Magna
St.","Petrolina","SC","28250"),
(13,"lobortis@rutm.com","Russell", "Wooten","5849 Et
Rd.","Monmouth","GA","465321"),
(14,"faucibus.leo.in@orciluctuset.co.uk","Jerry", "Hewitt","297-6603
Mauris. St.","Tourinne","TN","96217"),
(15,"lorem.sit@elementumduquis.edu","Jin", "Gordon","1079 Mus.
St.","Newtown","GA","6870");
INSERT INTO person
(PersonID,Email,FirstName,LastName,StreetAddress,City,State,Zipcode)
VALUES
(16,"eget.ipsum@ac.org","Zephania","Cain","195-1129 Augue. St.","Rod
Milici","NC","784512"),
(17,"magna@at.com","Lester", "Moses","Ap 144-4594 Tellus.
Rd.","Sinaai-Waas","NC","13036"),
(18,"gravida@consectetuer.ca","Burke", "Riggs","Ap 244-4588 Nec,
Road","Melipilla","SC","2923"),
```



(19,"sem.Pellentesque@orci.com","Zeus", "Gallegos", "Ap 265-7290 Molestie Ave", "Tywyn", "GA", "3132"),  
(20,"tellus.Aenean.egestas@turpisnec.net","Axel", "Dudley", "255-1145 Morbi Av.", "Montgomery", "TN", "86883306"),  
(21,"lacus.Quisque.purus@temporbibendum.org","Zeph", "Rhodes", "427 Arcu. Street", "Great Falls", "NC", "5130"),  
(22,"Fusce@tinciduntpede.co.uk","Emerson", "Underwood", "Ap 299-963 Gravida Avenue", "Acerra", "NC", "12195"),  
(23,"enim@malesuada.org","Tarik", "Bond", "P.O. Box 270, 8255 Mauris St.", "Denbigh", "SC", "3604534"),  
(24,"blandit.Nam.nulla@nequesedsem.co.uk","Brian", "Langley", "P.O. Box 623, 4260 Nascetur Av.", "Aisén", "GA", "2014"),  
(25,"Donec.tincidunt@sapienmolestieorci.ca","Jermaine", "Christensen", "P.O. Box 751, 7527 Lectus Rd.", "Bressoux", "GA", "698458"),  
(26,"euismod@Cras.com","Rafael", "Benson", "P.O. Box 610, 1276 Vulputate Avenue", "Whitehaven", "TN", "60917"),  
(27,"quam.elementum@nonluctussit.edu","Bruce", "Hurst", "Ap 931-1276 Nulla. St.", "Boston", "VA", "93045"),  
(28,"Morbi.sit@etlacinia.co.uk","Joseph", "Underwood", "7986 Ipsum Rd.", "San Donato di Ninea", "NC", "9336"),  
(29,"Curabitur@Donecnibh.net","Zachery", "Sexton", "586 Molestie Avenue", "Lowell", "VA", "86275"),  
(30,"euismod.mauris@Crasconvallisconvallis.net","Clarke", "Kramer", "Ap 653-4118 Iaculis Avenue", "Castel Giorgio", "VA", "2629");  
INSERT INTO person  
(PersonID,Email,FirstName,LastName,StreetAddress,City,State,Zipcode)  
VALUES  
(31,"velit.Pellentesque.ultricies@ametmassa.edu","Zeus", "Carrillo", "Ap 231-6600 Eget Av.", "Nurallao", "NC", "456321"),  
(32,"lectus.rutrum@amet.edu","Curran", "Carlson", "P.O. Box 289, 1527 Dui, Ave", "Melle", "NC", "8022"),  
(33,"ut.nisi@Donecfelis.edu","Julian", "Noble", "728-1852 Risus. Rd.", "Zaragoza", "TN", "29479"),  
(34,"dui.Cras@congue.co.uk","Galvin", "Washington", "P.O. Box 458, 2438 Ligula. Rd.", "Joliet", "SC", "79863"),  
(35,"ut@mipedenonummy.co.uk","Addison", "Farmer", "7751 Nisi Rd.", "Zonhoven", "VA", "042207"),  
(36,"non.enim.Mauris@fringilla.net","Gareth", "Reese", "P.O. Box 994, 3155 Et Road", "Albano di Lucania", "NC", "52804"),  
(37,"sem.Nulla.interdum@semutdolor.co.uk","Martin", "Wall", "P.O. Box 189, 6959 Non, Av.", "Vilna", "VA", "40832"),  
(38,"Maecenas.ornare@sapienimperdietornare.edu","Hyatt", "Justice", "Ap 747-2278 Eu Rd.", "Massello", "NC", "105165"),  
(39,"tellus@molestie.com","Jerome", "Mendez", "Ap 711-9315 Et, St.", "Dalcahue", "SC", "69649"),



```
(40,"quis@euneque.net","Lane", "Trevino", "5511 Nascetur  
St.", "Alwar", "NC", "66405"),  
(41,"eu.enim@adipiscingelit.org", "Rudyard", "Montgomery", "P.O. Box 298,  
5379 Eget, Av.", "Arquata del Tronto", "TN", "10619"),  
(42,"fringilla.porttitor@egestasblanditNam.org", "Fitzgerald",  
"Ellis", "3299 Elementum Road", "Ilkeston", "GA", "2233598"),  
(43,"dolor@magna.net", "Murphy", "Wiley", "Ap 423-5099 Laoreet,  
Rd.", "Hamoir", "VA", "456321"),  
(44,"aliquet@Fuscefeugiat.edu", "Zephania", "Wallace", "783-5420 Ut  
Rd.", "Roccamena", "SC", "63264"),  
(45,"at.sem@temporaugueac.com", "Simon", "Ramirez", "753-8739 Ullamcorper,  
St.", "Dégelis", "NC", "54239");
```

```
INSERT INTO Chef (Specialty, Salary, FKPersonID)  
VALUES  
("Lamb", 45000, 1),  
("Filet Mignon", 62000, 2),  
("Chicken", 45000, 3),  
("Fish", 56000, 4),  
("Steak", 60000, 5),  
("Pasta", 4500, 6),  
("Hibachi", 40000, 7),  
("Desert", 39000, 8),  
("Noodles", 35000, 9),  
("Cake", 40000, 10),  
("Ribs", 45000, 11),  
("Salmon", 47000, 12),  
("Stirfry", 40000, 13),  
("Sandwich", 35000, 14),  
("Pork", 45000, 15);
```

```
INSERT INTO Student (Major, ExpectedGraduationDate, FKPersonID)  
VALUES  
("Fish", "2020-06-28", 15),  
("Dessert", "2021-06-28", 16),  
("Pasta", "2020-12-28", 17),  
("Meat", "2019-12-28", 18),  
("Fish", "2020-06-28", 19),  
("Meat", "2020-12-28", 20),  
("Pasta", "2021-06-28", 21),
```



```

("Dessert", "2021-12-28",22),
("Dessert", "2021-12-28",23),
("Dessert", "2020-06-28",24),
("Dessert", "2020-06-28",25),
("Meat", "2019-12-28",26),
("Meat", "2019-12-28",27),
("Meat", "2020-06-28",28),
("Meat", "2020-06-28",29),
("Meat", "2021-06-28",30),
("Meat", "2021-06-28",31),
("Fish", "2022-06-28",32),
("Fish", "2020-06-28",33),
("Fish", "2020-06-28",34),
("Fish", "2020-06-28",35),
("Fish", "2020-06-28",36),
("Pasta", "2020-06-28",37),
("Pasta", "2020-06-28",38),
("Pasta", "2020-06-28",39),
("Pasta", "2020-12-28",40),
("Dessert", "2021-12-28",41),
("Meat", "2021-12-28",41),
("Fish", "2022-06-28",42),
("Pasta", "2022-06-28",43),
("Pasta", "2022-06-28",44),
("Pasta", "2022-06-28",45);

```

```

INSERT INTO recipe (RecipeID, Name, Downloads, PublicationDate, Category,
Cuisine, NumberServed) VALUES
("1", "bacon burger","15782", "2020-10-23", "Burger", "American", 1),
("2", "falafel gyro","651", "2020-01-01", "Pita", "mediterranean", 1),
("3", "pad thai","78912", "2019-10-14", "Noodles", "thai", 2),
("4", "bbq sandwich","12688", "2019-12-08", "BBQ", "American", 1),
("5", "fried chicken","789654", "2019-07-22", "Chicken", "American", 2),
("6", "teriyaki wings","456", "2019-11-06", "Wings", "American", 2),
("7", "spinach dip","24862", "2019-04-25", "Veggies", "American", 4),
("8", "fruit smoothie","98561", "2020-06-09", "Fruit", "American", 1),
("9", "Chicken noodle soup","7895", "2019-03-08", "Soup", "American", 3),
("10", "Sub Sandwich","14526", "2020-11-02", "Sandwich", "American", 1),
("11", "Coconut Curry", "275", "2006-12-8", "Spicy", "Indian", 2),
("12", "Chocolate Pie", "12548", "2012-5-7", "Dessert", "American", 4),
("13", "Fried Rice Stir Fry", 275, "2009-1-18", "Rice", "Chinese", 6),
("14", "Caesar Salad", 275, "2016-9-4", "Salad", "American", 4),
("15", "Cheesy Calzone", 275, "2033-6-7", "Calzone", "italian", 2),
("16", "Deep Dish", 275, "201-5-30", "Pizza", "italian", 8),

```



```
(""17", "Apples & Caramel", 275, "2014-11-9", "Snack", "American", 1),  
("18", "Mac and Cheese", 275, "2015-7-25", "Sides", "American", 6),  
("19", "Rainbow Roll", 275, "2018-2-25", "Sushi", "japanese", 3),  
("20", "Nachos", 275, "2019-11-15", "Chip", "Mexican", 2);
```

```
INSERT INTO DevelopmentProject (Name)  
Values  
("Cake"),  
("Pot Roast"),  
("Pork Chop"),  
("Steak"),  
("Ribs"),  
("Pasta Salad"),  
("Salmon"),  
("Meat loaf"),  
("Fudge"),  
("Cake Pops"),  
("SwordFish"),  
("Lobster"),  
("Crab Legs"),  
("Mussels"),  
("Sausage"),  
("Chicken Pot Pie"),  
("Beef Stew"),  
("Cupcakes"),  
("Ice Cream"),  
("Brownies"),  
("Shrimp Scampi");
```

```
INSERT INTO eCookBook (NumberOfRecipes, Theme, Name)  
VALUES  
(3, "Dessert", "Simple Desserts"),  
(4, "Pasta", "Simple Pasta"),  
(4, "Meat", "Tasty Meat"),  
(2, "Fish", "Easy Fish");
```



```
INSERT INTO SpecialtyEquipment (Name)
Value
("cast iron pan"),
("Blender"),
("Grill"),
("Brush"),
("Bag"),
("Mixer"),
("Spatula"),
("Skillet"),
("Pressure Cooker"),
("Large Pot");
```

```
INSERT INTO Award (Name)
VALUES
("Best Taste"),
("Prettiest Presentation"),
("Best Flavor"),
("Most Healthy"),
("Most Creative"),
("Greatest use of Spices"),
("Best Looking"),
("Quickest Prep"),
("Best Colors"),
("Most Inspired");
```

```
INSERT INTO Recipe_Award (RecipeID,AwardID,Date)
Values
(13,1,"2020-10-23"),
(2,3,"2020-01-01"),
(3,2,"2019-10-14"),
(4,5,"2019-12-08"),
(15,4,"2019-07-22"),
(6,7,"2019-11-06"),
(7,6,"2019-04-25"),
(8,9,"2020-06-09"),
(13,1,"2019-03-08"),
(9,8,"2018-04-22"),
(13,10,"2020-11-02");
```



```
INSERT INTO Instruction (Instruction)
VALUE
("Crack an egg."),
("Empty contents of egg into skillet."),
("Allow time to pass."),
("Eat egg."),
("non, dapibus rutrum, justo. Praesent luctus."),
("Curabitur egestas nunc sed"),
("quis, tristique ac, eleifend vitae, erat. Vivamus nisi. Mauris nulla."),
("dolor. Nulla semper tellus id nunc interdum feugiat."),
("dictum cursus. Nunc mauris elit, dictum eu, eleifend nec, malesuada"),
("Vivamus sit amet risus. Donec egestas. Aliquam nec enim. Nunc"),
("lectus. Cum sociis natoque penatibus et magnis dis parturient montes,"),
("ut lacus. Nulla tincidunt, neque vitae semper egestas, urna justo"),
("justo sit amet nulla. Donec non justo. Proin non massa"),
("interdum ligula eu enim. Etiam imperdiet dictum magna. Ut tincidunt"),
("libero est, congue a, aliquet vel, vulputate eu, odio. Phasellus"),
("fames ac turpis egestas. Aliquam fringilla cursus purus. Nullam
scelerisque"),
("sagittis lobortis mauris. Suspendisse aliquet molestie tellus. Aenean
egestas hendrerit"),
("vitae odio sagittis semper. Nam tempor diam dictum sapien. Aenean"),
("sapien, cursus in, hendrerit consectetur, cursus et, magna. Praesent
interdum"),
("in consequat enim diam vel arcu. Curabitur ut odio vel"),
("Maecenas ornare egestas ligula. Nullam feugiat placerat velit. Quisque
varius."),
("sed dolor. Fusce mi lorem, vehicula et, rutrum eu, ultrices"),
("at lacus. Quisque purus sapien, gravida non, sollicitudin a,
malesuada"),
("sed turpis nec mauris blandit mattis. Cras eget nisi dictum"),
("luctus. Curabitur egestas nunc sed libero. Proin sed turpis nec");
```



```
INSERT INTO Ingredient (Units, Name)
VALUES
("Ounces", "Chicken"), ("Ounces", "Steak"), ("Ounces", "Salmon"),
("Whole", "Egg"), ("Cups", "Rice"), ("Teaspoons", "Garlic"),
("Cups", "Peanuts"), ("Cups", "Noodles"), ("Cups", "Beans"),
("Whole", "Tomato");
INSERT INTO Ingredient(Units,Name)
VALUES
("Teaspoons", "Salt"),
("Cups", "Cheese"),
("Teaspoons", "Pepper"),
("Whole", "Buns"),
("Whole", "Bacon"),
("Ounces", "Hamburger"),
("Whole", "Pita Roll"),
("Cups", "Shredded Lettuce"),
("Cups", "Onion"),
("Cups", "Thai Noodles"),
("Cups", "Thai Sauce"),
("Cups", "Green Pepper"),
("Ounces", "Pork"),
("Cups", "Barbecue Sauce"),
("Cups", "Milk"),
("Cups", "Flour"),
("Whole", "Chicken Wings"),
("Teaspoon", "Sugar"),
("Cups", "Spinach"),
("Teaspoons", "Spice"),
("Whole", "Apple"),
("Whole", "Banana"),
("Cups", "Chicken Broth"),
("Cups", "Carrots"),
("Whole", "Sub Roll"),
("Cups", "Ham"),
("Cups", "Turkey"),
("Whole", "Coconut"),
("Teaspoon", "Curry Sauce"),
("Cups", "Coconut Milk"),
("Whole", "Pizza Dough"),
("Cups", "Caramel"),
("Cups", "Elbow Pasta"),
("Cups", "Nacho Chips");
```



```
INSERT INTO Recipe_Chef (RecipeID,ChefID)
Values
(1,15),(2,14),(3,13),(4,12),
(5,11),(6,10),(7,9),(8,7),
(9,8),(10,6),(11,5),
(12,4),(13,3),
(14,2),(15,1),(1,10),(1,3),
(1,13),(2,3),(2,4),(3,3),(16,1),(16,4),(17,9),(18,10),(19,15),(20,7);
```

```
INSERT INTO Recipe_eCookBook(eCookBookID,RecipeID)
VALUES
(1,8),(1,12),(1,17),(2,3),(2,9),(2,15),(2,18),(3,4),(3,5),(3,6),(3,1),(4,1
9),(4,2);
```

```
INSERT INTO Recipe_Ingredient(RecipeID,IngredientID,Quantity)
VALUES
(1,11,1),(1,13,1),(1,4,1),(1,16,8),
(1,14,1),(2,11,1),(2,13,1),(2,6,1),
(2,1,8),(2,17,1),(3,11,1),(3,13,1),
(3,21,1),(3,20,2),(3,19,1),(4,23,16),
(4,13,1),(4,14,1),(4,24,1),(4,11,1),(5,1,16),(5,6,1),(5,13,1),(5,12,1),(5,
26,1),(6,27,6),(6,13,1),(6,11,1),(6,30,1),(6,24,1),(7,29,1),(7,12,1),(7,13
,1),(7,11,1),(7,18,1),(8,28,2),(8,25,1),(8,31,1),(8,32,1),(8,34,1),(9,1,6)
,(9,33,1),(9,13,1),(9,11,1),(9,8,1),(10,35,1),(10,10,1),(10,41,1),(10,12,1
),(10,6,1);
```

```
INSERT INTO Recipe_Ingredient(RecipeID,IngredientID,Quantity)
VALUES
(11,30,1),(11,40,1),(11,39,1),(11,11,1),(11,13,1),
(12,42,1),(12,26,1),
(12,28,2),(12,4,1),(12,25,1),(13,9,1),(13,5,1),
(13,18,1),(13,22,1),(13,30,1),(14,18,1),(14,10,1),(14,13,1),
(14,11,1),(14,12,1),(15,36,4),(15,12,1),(15,41,1),
(15,19,1),(15,37,10),(16,41,1),
(16,12,1),(16,19,1),(16,10,1),(16,6,1),(17,31,1),(17,42,1),
(17,28,1),(17,25,1),(17,7,1),
(18,43,1),(18,12,1),
(18,25,1),(18,11,1),(18,15,2),(19,3,8),
(19,5,1),(19,12,1),(19,11,1),(19,13,1),
(20,44,1),(20,12,1),(20,22,1),(20,19,1),(20,18,1);
```



```
INSERT INTO Recipe_Instruction(RecipeID, InstructionStepID)
Values
(1,1), (1,2), (1,3), (1,4), (2,5), (3,6),
(4,7), (5,8), (6,9), (7,10), (8,11),
(9,12), (10,13), (11,14), (12,15),
(13,16), (14,17), (15,18),
(16,19), (17,20), (18,21), (19,22), (20,23), (20,24);
```

```
INSERT INTO Recipe_SpecialtyEquipment(RecipeID, EquipmentID)
VALUES
(1,1), (7,2), (8,2), (11,2), (1,3), (4,3), (6,3),
(17,4), (6,4), (5,5), (12,6), (13,7),
(18,7), (11,8), (13,8), (3,8), (9,9), (18,10), (19,10), (20,10);
```

```
INSERT INTO LeadChef(FKLeadChefID)
VALUES
(3), (10), (3), (13),
(12), (11), (10), (9),
(7), (8), (6), (5), (4), (3),
(2), (1), (1), (9), (10),
(15), (7);
```

```
INSERT INTO Recipe_LeadChef(RecipeID, LeadChefID)
VALUES
(1,1), (2,2), (3,3), (4,4),
(5,5), (6,6),
(7,7), (8,8),
(9,9), (10,10), (11,11),
(12,12), (13,13),
(14,14), (15,15),
(16,16), (17,17),
(18,18), (19,19), (20,20);
```

```
INSERT INTO Recipe_Student(RecipeID, StudentID)
Values
(13,27), (19,31), (17,23), (11,16), (16,30), (2,15), (10,1), (16,22), (8,10), (5,1)
,(4,3), (6,4),
(9,13), (14,16), (5,15), (17,7), (14,19), (7,13), (18,32), (7,1), (7,15), (7,16), (1
7,14),
(10,15), (1,8), (12,2), (6,18), (4,28), (14,15), (3,10), (7,3), (5,29), (5,20), (5,5
),(13,21), (15,26), (1,21), (19,26), (6,1), (5,27), (12,28), (1,1), (9,15), (9,9), (9
,32),
(4,18), (16,7), (4,20), (7,27), (3,30), (15,31), (16,19), (18,21), (15,17), (13,11
),(3,22), (5,31), (18,14),
```



```
(10,10), (4,1), (10,23), (12,11), (5,24), (5,9), (5,16), (7,18), (6,23), (11,11), (8  
,30), (16,4), (12,24),  
(14,14), (9,27), (19,16), (18,31), (16,6), (17,1), (10,3), (11,9), (4,21), (9,14), (  
3,28), (2,11), (8,17), (16,5),  
(4,2), (10,5), (8,3), (17,28), (1,12), (8,29), (13,24), (7,12);
```

```
INSERT INTO Chef_DevelopmentProject(ChefID, ProjectID)  
Values
```

```
(12,11), (12,10), (12,20), (3,7), (1,16), (12,16), (14,17), (11,15), (7,5), (6,13),  
(9,6), (3,12), (2,5), (1,15), (4,1), (11,13), (9,12), (2,3), (5,6), (6,19), (13,4),  
(11,20), (9,2), (14,18), (5,8), (2,9), (10,14), (7,8), (2,2);
```

```
INSERT INTO LeadChef_DevelopmentProject(LeadChefID, ProjectID)  
Values  
(4,1), (2,2), (2,3), (13,4), (7,5), (5,6),  
(3,7), (5,8), (2,9), (12,10),  
(9,12), (6,13), (10,14), (1,15),  
(11,15), (12,16), (14,17),  
(6,19), (11,20);
```

```
INSERT INTO Student_DevelopmentProject(StudentID, ProjectID)  
VALUES  
(24,2), (31,4), (13,12), (19,6), (13,11), (22,18), (5,9), (31,3), (13,9), (12,19), (1  
,13),  
(6,10), (25,4), (19,7), (32,12), (2,16), (5,7), (7,19), (20,1), (22,4), (9,9), (25,1  
7), (1,16),  
(21,7), (7,1), (15,12), (28,11), (3,7), (5,16), (1,7), (2,15), (18,10), (23,13), (16  
,8), (6,2), (17,19),  
(17,17), (15,7), (20,8), (29,14), (21,17), (10,16), (21,18), (4,17), (18,19), (8,5)  
, (30,19),  
(9,8), (20,18), (7,9), (7,14), (25,10), (5,1), (30,1), (30,3), (20,17), (27,13), (12  
,16),  
(11,15), (21,5), (16,10), (8,9), (17,15), (29,6), (14,5), (24,9), (5,2), (12,3), (21  
,4), (28,4), (8,13),  
(14,19), (21,10), (8,1), (31,16), (8,12), (28,2), (9,17), (28,14), (7,4), (26,9), (1  
6,6),  
(2,2), (9,3), (22,1), (21,6), (5,5), (28,1), (14,4), (16,18), (26,8);
```



# Report Queries

## Chef Reports:

- a) Listing unpublished recipe projects by lead chef

The screenshot shows a database interface with several tabs at the top: chef, chef, person, developmentproject (which is selected), leadchef, leadchef\_developmentproject, and developmentproject. Below the tabs is a toolbar with various icons. The main area contains a numbered SQL query:

```
1 •   SELECT Person.FirstName,Person.LastName,DevelopmentProject.ProjectID, DevelopmentProject.name
2   Ⓜ FROM (((Person
3     Ⓛ INNER JOIN Chef ON Person.PersonID = Chef.FKPersonID)
4     Ⓛ INNER JOIN LeadChef ON Chef.ChefID = LeadChef.FKLeadChefID)
5     Ⓛ INNER JOIN LeadChef_DevelopmentProject ON LeadChef.LeadChefID = LeadChef_DevelopmentProject.LeadChefID)
6     Ⓛ INNER JOIN DevelopmentProject ON LeadChef_DevelopmentProject.ProjectID = DevelopmentProject.ProjectID
7   ORDER BY LastName;
8
```

Below the query is a "Result Grid" table with four columns: FirstName, LastName, ProjectID, and name. The data is as follows:

	FirstName	LastName	ProjectID	name
▶	Jackson	Ashley	13	Crab Legs
	Jackson	Ashley	19	Ice Cream
	Merritt	Grant	15	Sausage
	Merritt	Grant	20	Brownies
	Nathan	Kaufman	2	Pot Roast
	Nathan	Kaufman	3	Pork Chop
	Nathan	Kaufman	9	Fudge
	Nathan	Kaufman	5	Ribs
	Aquila	Kerr	4	Steak
	Andrew	Massey	12	Lobster
	Vaughan	Obrien	14	Mussels
	Reece	Parilla	6	Paella Salad

At the bottom left, it says "Result 10".



- b) Listing published recipe projects by Lead chef

The screenshot shows a database interface with a query editor at the top and a result grid below. The query editor contains the following SQL code:

```
1 •  SELECT Person.FirstName,Person.LastName,Recipe.RecipeID, Recipe.name
2   FROM (((Person
3     INNER JOIN Chef ON Person.PersonID = Chef.FKPersonID)
4     INNER JOIN LeadChef ON Chef.ChefID = LeadChef.FKLeadChefID)
5       INNER JOIN Recipe_LeadChef on LeadChef.LeadChefID = Recipe_LeadChef.LeadChefID)
6         INNER JOIN Recipe on Recipe_LeadChef.RecipeID = Recipe_LeadChef.RecipeID
7           ORDER BY LastName;
8
```

The result grid displays the following data:

	FirstName	LastName	RecipeID	name
▶	Jackson	Ashley	10	Sub Sandwich
	Jackson	Ashley	2	falafel gyro
	Jackson	Ashley	16	Deep Dish
	Jackson	Ashley	6	teriyaki wings
	Jackson	Ashley	5	fried chicken
	Jackson	Ashley	15	Cheesy Calzone
	Jackson	Ashley	9	Chicken noodle soup
	Jackson	Ashley	8	fruit smoothie
	Jackson	Ashley	3	pad thai
	Jackson	Ashley	11	Coconut Curry
	Jackson	Ashley	18	Mac and Cheese
	Jackson	Ashley	20	Narhne

Result 11 ×



c) Listing all recipe projects and included assigned chefs

The screenshot shows a database interface with a query editor and a result grid.

**Query Editor:**

```
1  SELECT Person.FirstName,Person.LastName,Recipe.RecipeID,Recipe.Name,DevelopmentProject.ProjectID, DevelopmentProject.Name
2  FROM (((Person
3    INNER JOIN Chef ON Person.PersonID = Chef.FKPersonID)
4    INNER JOIN Recipe_Chef ON Chef.ChefID = Recipe_Chef.ChefID)
5    INNER JOIN Recipe ON Recipe_Chef.RecipeID = Recipe.RecipeID)
6    INNER JOIN Chef_DevelopmentProject ON Chef.ChefID = Chef_DevelopmentProject.ChefID)
7    INNER JOIN DevelopmentProject ON Chef_DevelopmentProject.ProjectID = DevelopmentProject.ProjectID
8  ORDER BY LASTNAME;
```

**Result Grid:**

FirstName	LastName	RecipeID	Name	ProjectID	Name
Jackson	Ashley	5	fried chicken	20	Brownies
Jackson	Ashley	5	fried chicken	15	Sausage
Jackson	Ashley	5	fried chicken	13	Crab Legs
Alvin	Brennan	14	Caesar Salad	3	Pork Chop
Alvin	Brennan	14	Caesar Salad	2	Pot Roast
Alvin	Brennan	14	Caesar Salad	5	Ribs
Alvin	Brennan	14	Caesar Salad	9	Fudge
Connor	English	16	Deep Dish	15	Sausage
Connor	English	15	Cheesy Calzone	16	Chicken Pot Pie
Connor	English	16	Deep Dish	16	Chicken Pot Pie
Connor	English	15	Cheesy Calzone	15	Sausage
Merritt	Grant	10	Sub Sandwich	13	Crab Legs



## Student Report:

- a) Listing all students and their published recipe projects with assigned chefs.

The screenshot shows a database interface with a query editor and a results grid. The query editor contains the following SQL code:

```
1  SELECT Person.FirstName, Person.LastName, Student.StudentID, Recipe.Name, Recipe_Chef.ChefID
2  FROM (((Person
3    INNER JOIN Student ON Person.PersonID = Student.FKPersonID)
4    INNER JOIN Recipe_Student ON Student.StudentID = Recipe_Student.StudentID)
5    INNER JOIN Recipe ON Recipe_Student.RecipeID = Recipe.RecipeID)
6    INNER JOIN Recipe_Chef On Recipe.RecipeID = Recipe_Chef.RecipeID
7  ORDER BY LASTNAME;
```

The results grid displays the following data:

	FirstName	LastName	StudentID	Name	ChefID
▶	Rafael	Benson	12	bacon burger	3
	Rafael	Benson	12	bacon burger	10
	Rafael	Benson	12	bacon burger	13
	Rafael	Benson	12	bacon burger	15
	Rafael	Benson	12	spinach dip	9
	Tarik	Bond	9	Coconut Curry	5
	Tarik	Bond	9	Chicken noodle soup	8
	Tarik	Bond	9	fried chicken	11
	Zephania	Cain	2	Chocolate Pie	4
	Zephania	Cain	2	bbq sandwich	12
	Curran	Carlson	18	spinach dip	9
	Curran	Carlson	18	hhn sandwirth	12

Result 11 ×



## Management Report:

- a) Listing all chefs including names and salaries.

```
1  SELECT Person.FirstName,Person.LastName,Chef.Salary
2  From Chef
3  INNER JOIN Person
4  ON Chef.FKPersonID = Person.PersonID;
```

Result Grid | Filter Rows:  Export: Wrap Cell Content

	FirstName	LastName	Salary
▶	Connor	English	45000
	Alvin	Brennan	62000
	Reece	Tyler	45000
	Aquila	Kerr	56000
	Troy	Reeves	60000
	Merritt	Grant	4500
	Andrew	Massey	40000
	Vaughan	Obrien	39000
	Alvin	Little	35000
	Nathan	Kaufman	40000
	Jackson	Ashley	45000
▶	Darren	Darrell	47000

Result 3 ×



- b) Listing all students with names, a count of assigned projects, unpublished recipe projects, and expected graduation date.

SQL File 13\* × student\_developmentproject recipe\_student

```
1 •  SELECT Person.FirstName,Person.LastName,Student.StudentID,
2     COUNT(Recipe_Student.StudentID),COUNT(Student_DevelopmentProject.StudentID),
3     Student.ExpectedGraduationDate
4   FROM (((Person
5     INNER JOIN Student ON Person.PersonID = Student.FKPersonID)
6     INNER JOIN Recipe_Student ON Student.StudentID = Recipe_Student.StudentID)
7     INNER JOIN Student_DevelopmentProject ON Student.StudentID = Student_DevelopmentProject.StudentID)
8   Group by StudentID;
9
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

FirstName	LastName	StudentID	COUNT(Recipe_Student.StudentID)	COUNT(Student_DevelopmentProject.StudentID)	ExpectedGraduation
Jin	Gordon	1	21	21	2020-06-28
Zephania	Cain	2	6	6	2021-06-28
Lester	Moses	3	4	4	2020-12-28
Burke	Riggs	4	2	2	2019-12-28
Zeus	Gallegos	5	18	18	2020-06-28
Axel	Dudley	6	2	2	2020-12-28
Zeph	Rhodes	7	10	10	2021-06-28
Emerson	Underwood	8	5	5	2021-12-28
Tarik	Bond	9	12	12	2021-12-28
Brian	Langley	10	3	3	2020-06-28
Jermaine	Christensen	11	4	4	2020-06-28
Rafael	Benson	12	6	6	2019-12-28

Result 5 ×



# Recipes and eCookBooks:

- a) List each recipe with recipe name, lead chef, and number of times downloaded.

The screenshot shows a database interface with a query editor and a result grid. The query editor contains the following SQL code:

```
leadchef x recipe_chef chef_developmentproject student recipe
1 •  SELECT Recipe.Name, Person.FirstName, Person.LastName, Recipe_LeadChef.LeadChefID, Recipe.Downloads
2   From (((Person
3     INNER JOIN Chef ON Person.PersonID = Chef.FKPersonID)
4     INNER JOIN LeadChef ON Chef.ChefID = LeadChef.FKLeadChefID)
5     INNER JOIN Recipe_LeadChef on LeadChef.LeadChefID = Recipe_LeadChef.LeadChefID)
6     INNER JOIN Recipe on Recipe_LeadChef.RecipeID = Recipe_LeadChef.RecipeID);
7
```

The result grid displays the following data:

	Name	FirstName	LastName	LeadChefID	Downloads
▶	bacon burger	Reece	Tyler	1	15782
	bacon burger	Nathan	Kaufman	2	15782
	bacon burger	Reece	Tyler	3	15782
	bacon burger	Russell	Wooten	4	15782
	bacon burger	Reece	Padilla	5	15782
	bacon burger	Jackson	Ashley	6	15782
	bacon burger	Nathan	Kaufman	7	15782
	bacon burger	Alvin	Little	8	15782
	bacon burger	Andrew	Massey	9	15782
	bacon burger	Vaughan	Obrien	10	15782
	bacon burger	Merritt	Grant	11	15782
	bacon burger	Troy	Reeves	12	15782
	bacon burger	Aquila	Kerr	13	15782

Result 3 x



- b) List each recipe name with its ingredients. Include Quantity and Units.

The screenshot shows a MySQL Workbench interface. The top window is the SQL editor, displaying the following query:

```
1 •  SELECT Recipe.Name,Ingredient.Name, Recipe_Ingredient.Quantity
2   From ((Recipe
3     INNER JOIN Recipe_Ingredient ON Recipe.RecipeID = Recipe_Ingredient.RecipeID)
4     INNER JOIN Ingredient ON Recipe_Ingredient.IngredientID = Ingredient.IngredientID);
5
```

The bottom window is the Result Grid, showing the output of the query:

Name	Name	Quantity
bacon burger	Egg	1
bacon burger	Salt	1
bacon burger	Pepper	1
bacon burger	Buns	1
bacon burger	Hamburger	8
falafel gyro	Chicken	8
falafel gyro	Garlic	1
falafel gyro	Salt	1
falafel gyro	Pepper	1
falafel gyro	Pita Roll	1



c) List each recipe name with its instructions.

The screenshot shows a database interface with a query editor and a result grid. The query editor contains the following SQL code:

```
1 •  SELECT Recipe.Name, Instruction.InstructionStepID, Instruction.Instruction
2  From ((Recipe
3    INNER JOIN Recipe_Instruction ON Recipe.RecipeID = Recipe_Instruction.RecipeID)
4    INNER JOIN Instruction ON Recipe_Instruction.InstructionStepID = Instruction.InstructionStepID);
5
```

The result grid displays the following data:

Name	InstructionStepID	Instruction
bacon burger	1	Crack an egg.
bacon burger	2	Empty contents of egg into skillet.
bacon burger	3	Allow time to pass.
bacon burger	4	Eat egg.
falafel gyro	5	non, dapibus rutrum, justo. Praesent luctus.
pad thai	6	Curabitur egestas nunc sed quis, tristique ac, eleifend vitae, erat. Vivamus ...
bbq sandwich	7	dolor. Nulla semper tellus id nunc interdum feugi...
fried chicken	8	dictum cursus. Nunc mauris elit, dictum eu, eleif...
teriyaki wings	9	Vivamus sit amet risus. Donec egestas. Aliquam ...
spinach dip	10	lectus. Cum sociis natoque penatibus et magnis ...
fruit smoothie	11	ut lacus. Nulla tincidunt, neque vitae semper eg...
Chicken noo...	12	justo sit amet nulla. Donec non justo. Proin non ...
Sub Sandwich	13	

Result 5



- d) List each eCookBook with eCookBook name and number of recipes in it.

The screenshot shows the MySQL Workbench interface. At the top, there are tabs for 'leadchef' (selected), 'recipe\_chef', 'chef\_developmentproject', 'student', 'recipe', and 'recipe\_in'. Below the tabs is a toolbar with various icons. A query editor window contains the following SQL code:

```
1 •  SELECT eCookBook.name, eCookBook.NumberOfRecipes
2   From eCookBook;
3
```

Below the query editor is a results grid titled 'Result Grid' with columns 'name' and 'NumberOfRecipes'. The data is as follows:

	name	NumberOfRecipes
▶	Simple Desserts	3
	Simple Pasta	4
	Tasty Meat	4
	Easy Fish	2



## Awards:

a) List all recipes that have received awards including the name of the award. Once with a natural join and once with a join statement.

The screenshot shows a database interface with four tabs at the top: 'cntr' (disabled), 'award' (selected), 'recipe', and 'recipe\_award'. Below the tabs is a toolbar with icons for file operations, search, and navigation. A status bar at the bottom right says 'Limit to 1000 row'. The main area contains a numbered SQL query:

```
1 •   SELECT Award.Name, Recipe_Award.RecipeID  
2   FROM Award  
3   NATURAL JOIN Recipe_Award);  
4  
5
```

The screenshot shows a 'Result Grid' with columns 'Name' and 'RecipeID'. The data is as follows:

	Name	RecipeID
▶	Best Flavor	2
	Prettiest Presentation	3
	Most Creative	4
	Best Looking	6
	Greatest use of Spi...	7
	Best Colors	8
	Quickest Prep	9
	Best Taste	13



```
1 •  SELECT Award.Name,Recipe_Award.RecipeID  
2   FROM (Award  
3   INNER JOIN Recipe_Award ON Award.AwardID = Recipe_Award.AwardID);  
4  
5
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	Name	RecipeID		
▶	Best Flavor	2		
	Prettiest Presentation	3		
	Most Creative	4		
	Best Looking	6		
	Greatest use of Spi...	7		
	Best Colors	8		
	Quickest Prep	9		
	Best Taste	13		
		--		



## Stored Procedure:

The screenshot shows a MySQL Workbench interface with a query editor window. The tabs at the top are 'leadchef' (selected), 'recipe\_chef', 'chef\_developmentproject', and 'student'. Below the tabs is a toolbar with icons for file operations, search, and execution. A limit of '1000 rows' is set. The code in the editor is:

```
2      DELIMITER $$  
3  
4 •  CREATE PROCEDURE getChefIDWithHighSalary()  
5   BEGIN  
6       SELECT ChefID  
7       FROM Chef  
8       WHERE Chef.Salary > 50000;  
9   END $$  
10 • CALL getChefIDWithHighSalary();
```

Below the code is a result grid labeled 'Result Grid' showing the output:

ChefID
2
4
5

## Trigger:

```
DROP TRIGGER IF EXISTS DropStudent;  
CREATE TRIGGER DropStudent  
    BEFORE DELETE ON PERSON  
    FOR EACH ROW  
    DELETE Student FROM Student WHERE (Student.FKPersonID =  
Person.PersonID);
```

## Example Delete Query :

```
DELETE FROM Recipe_Delight.Recipe_Award  
WHERE (RecipeID = 9) AND AwardID = 8) AND Date = ("2018-04-22);
```



## Example Update Query :

```
UPDATE AWARD SET `Name` = 'Most Tasty'  
WHERE (AwardID = 1);
```

## Example of INDEX for a Report (Management) :

```
CREATE INDEX NameIndex ON Person(LastName, FirstName);
```

## Advanced SQL Features:

- Person to Chef and Student Specialization
- Trigger "DropStudent" to delete the person data when deleting a student.
- Stored Procedure "GetChefIDWithHighSalary" to return the ChefID's of Chefs making over \$50,000.



# Executive Summary

Our team has created a database- ReciepeDelight through mySQL. The purpose of our database is to be a site to be able to store information for recipes, awards, students, and chefs.

Information about people at RD Institute can be found, or instructions to create particular recipes. Through creating tables, implementing and then testing them we can designed and documented a database solution of RecipeDelight for the RD Institute. The prototype database is filled with test data to show how this design can work for you.

An ecookbook will be an efficient and effective way to grab yummy recipes with simple ingredients to make a good meal! The eCookBook is sorted by theme of the recipes inside and will show how many recipes are in it. Not only is our database a ecookbook but also a tracker of the students and chefs enrolled and registered. The database table values include but are not limited to Persons- name, email, city, Recipes- awards, name, ingredients. Both Student and Chef participation in the creation of recipes is tracked along with any awards that recipe has received. In our software ingredients quantities come in english units and can be manipulated to serve more or less people.

The trigger we have chosen to add will automatically drop the all personal information on students while not deleting any recipes they were a part of. The three query reports - chefs, students and a management report. Features like counting how many awards a recipe that a particular chef has made wins, and updating date of publishing are also included in our database to enhance information offered to a user and delete a student from the database once they have graduated they no longer need to be enrolled and should be removed to keep our system clean. All features are in hope to ease usage for chefs, students, and others.

# Data Dictionary





<u>Column Name</u>	<u>Data Type</u>	<u>Null</u>	<u>Primary Key</u>	<u>Foreign Key</u>	<u>Referenced Table/Column</u>	<u>Auto Increment</u>	<u>Default Value</u>	<u>Unique</u>	<u>Unsigned</u>
eCookBookID	int(10)		YES			YES	NULL	YES	YES
NumberOfRecipes	int(11)	YES					NULL		
Theme	varchar(64)	YES					NULL		
Name	varchar(64)	YES					NULL		
<b>Table Name :</b>	Ingredient								
<u>Column Name</u>	<u>Data Type</u>	<u>Null</u>	<u>Primary Key</u>	<u>Foreign Key</u>	<u>Referenced Table/Column</u>	<u>Auto Increment</u>	<u>Default Value</u>	<u>Unique</u>	<u>Unsigned</u>
IngredientID	int(10)		YES			YES	NULL	YES	YES
Units	varchar(64)	YES					NULL		
Name	varchar(64)	YES					NULL		
<b>Table Name :</b>	Instruction								
<u>Column Name</u>	<u>Data Type</u>	<u>Null</u>	<u>Primary Key</u>	<u>Foreign Key</u>	<u>Referenced Table/Column</u>	<u>Auto Increment</u>	<u>Default Value</u>	<u>Unique</u>	<u>Unsigned</u>
InstructionStepID	int(10)		YES			YES	NULL	YES	YES
Instruction	varchar(255)	YES					NULL		
<b>Table Name :</b>	LeadChef								
<u>Column Name</u>	<u>Data Type</u>	<u>Null</u>	<u>Primary Key</u>	<u>Foreign Key</u>	<u>Referenced Table/Column</u>	<u>Auto Increment</u>	<u>Default Value</u>	<u>Unique</u>	<u>Unsigned</u>
LeadChefID	int(10)		YES	YES	LeadChef(LeadChefID)		NULL	YES	YES
FKLeadChefID	int(10)		YES	YES	Chef(ChefID)		NULL	YES	YES
<b>Table Name :</b>	LeadChef_DevelopmentProject								
<u>Column Name</u>	<u>Data Type</u>	<u>Null</u>	<u>Primary Key</u>	<u>Foreign Key</u>	<u>Referenced Table/Column</u>	<u>Auto Increment</u>	<u>Default Value</u>	<u>Unique</u>	<u>Unsigned</u>
LeadChefID	int(10)		YES	YES	LeadChef(LeadChefID)		NULL	YES	YES
ProjectID	int(10)		YES	YES	DevelopmentProject(ProjectID)		NULL	YES	YES
<b>Table Name :</b>	Person								
<u>Column Name</u>	<u>Data Type</u>	<u>Null</u>	<u>Primary Key</u>	<u>Foreign Key</u>	<u>Referenced Table/Column</u>	<u>Auto Increment</u>	<u>Default Value</u>	<u>Unique</u>	<u>Unsigned</u>
PersonID	int(10)		YES			YES	NULL	YES	YES
Email	varchar(64)	YES					NULL		
FirstName	varchar(64)	YES					NULL		
LastName	varchar(64)	YES					NULL		

StreetAddress	varchar(64)	YES						NULL			
City	varchar(64)	YES						NULL			
State	varchar(2)	YES						NULL			
Zipcode	int(11)	YES						NULL			
<hr/>											
<b>Table Name :</b>	Recipe_Award										
<u>Column Name</u>	<u>Data Type</u>	<u>Null</u>	<u>Primary Key</u>	<u>Foreign Key</u>	<u>Referenced Table/Column</u>	<u>Auto Increment</u>	<u>Default Value</u>	<u>Unique</u>	<u>Unsigned</u>		
RecipeID	int(10)		YES	YES	Recipe(RecipeID)			NULL	YES	YES	
AwardID	int(10)		YES	YES	Award(AwardID)			NULL	YES	YES	
Date	date		YES					NULL	YES		
<hr/>											
<b>Table Name :</b>	Recipe_Chef										
<u>Column Name</u>	<u>Data Type</u>	<u>Null</u>	<u>Primary Key</u>	<u>Foreign Key</u>	<u>Referenced Table/Column</u>	<u>Auto Increment</u>	<u>Default Value</u>	<u>Unique</u>	<u>Unsigned</u>		
RecipeID	int(10)		YES	YES	Recipe(RecipeID)			NULL	YES	YES	
ChefID	int(10)		YES	YES	Chef(ChefID)			NULL	YES	YES	
<hr/>											
<b>Table Name :</b>	Recipe_eCookBook										
<u>Column Name</u>	<u>Data Type</u>	<u>Null</u>	<u>Primary Key</u>	<u>Foreign Key</u>	<u>Referenced Table/Column</u>	<u>Auto Increment</u>	<u>Default Value</u>	<u>Unique</u>	<u>Unsigned</u>		
RecipeID	int(10)		YES	YES	Recipe(RecipeID)			NULL	YES	YES	
eCookBookID	int(10)		YES	YES	eCookBook(eCookBookID)			NULL	YES	YES	
<hr/>											
<b>Table Name :</b>	Recipe_Ingredient										
<u>Column Name</u>	<u>Data Type</u>	<u>Null</u>	<u>Primary Key</u>	<u>Foreign Key</u>	<u>Referenced Table/Column</u>	<u>Auto Increment</u>	<u>Default Value</u>	<u>Unique</u>	<u>Unsigned</u>		
RecipeID	int(10)		YES	YES	Recipe(RecipeID)			NULL	YES	YES	
IngredientID	int(10)		YES	YES	Ingredient(IngredientID)			NULL	YES	YES	
Quantity	int(11)	YES						NULL	YES	YES	
<hr/>											
<b>Table Name :</b>	Recipe_Instruction										
<u>Column Name</u>	<u>Data Type</u>	<u>Null</u>	<u>Primary Key</u>	<u>Foreign Key</u>	<u>Referenced Table/Column</u>	<u>Auto Increment</u>	<u>Default Value</u>	<u>Unique</u>	<u>Unsigned</u>		
RecipeID	int(10)		YES	YES	Recipe(RecipeID)			NULL	YES	YES	
InstructionStepID	int(10)		YES	YES	Instruction(InstructionStepID)			NULL	YES	YES	

<b>Table Name :</b>	Recipe_LeadChef									
<u>Column Name</u>	<u>Data Type</u>	<u>Null</u>	<u>Primary Key</u>	<u>Foreign Key</u>	<u>Referenced Table/Column</u>	<u>Auto Incrementation</u>	<u>Default Value</u>	<u>Unique</u>	<u>Unsigned</u>	
RecipeID	int(10)		YES	YES	Recipe(RecipeID)			NULL	YES	
LeadChefID	int(10)		YES	YES	LeadChef(LeadChefID)			NULL	YES	
<b>Table Name :</b> Recipe_SpecialtyEquipment										
<u>Column Name</u>	<u>Data Type</u>	<u>Null</u>	<u>Primary Key</u>	<u>Foreign Key</u>	<u>Referenced Table/Column</u>	<u>Auto Incrementation</u>	<u>Default Value</u>	<u>Unique</u>	<u>Unsigned</u>	
RecipeID	int(10)		YES	YES	Recipe(RecipeID)			NULL	YES	YES
EquipmentID	int(10)		YES	YES	SpecialtyEquipment(EquipmentID)			NULL	YES	YES
<b>Table Name :</b> Recipe_Student										
<u>Column Name</u>	<u>Data Type</u>	<u>Null</u>	<u>Primary Key</u>	<u>Foreign Key</u>	<u>Referenced Table/Column</u>	<u>Auto Incrementation</u>	<u>Default Value</u>	<u>Unique</u>	<u>Unsigned</u>	
RecipeID	int(10)		YES	YES	Recipe(RecipeID)			NULL	YES	YES
StudentID	int(10)		YES	YES	Student(StudentID)			NULL	YES	YES
<b>Table Name :</b> SpecialtyEquipment										
<u>Column Name</u>	<u>Data Type</u>	<u>Null</u>	<u>Primary Key</u>	<u>Foreign Key</u>	<u>Referenced Table/Column</u>	<u>Auto Incrementation</u>	<u>Default Value</u>	<u>Unique</u>	<u>Unsigned</u>	
EquipmentID	int(10)		YES				YES	NULL	YES	YES
Name	varchar(64)	YES						NULL		
<b>Table Name :</b> Student										
<u>Column Name</u>	<u>Data Type</u>	<u>Null</u>	<u>Primary Key</u>	<u>Foreign Key</u>	<u>Referenced Table/Column</u>	<u>Auto Incrementation</u>	<u>Default Value</u>	<u>Unique</u>	<u>Unsigned</u>	
StudentID	int(10)		YES				YES	NULL	YES	YES
Major	varchar(64)	YES						NULL		
ExpectedGraduationDate	date	YES						NULL		
FKPersonID	int(10)				YES	Person(PersonID)			NULL	YES
<b>Table Name :</b> Student_DevelopmentProject										
<u>Column Name</u>	<u>Data Type</u>	<u>Null</u>	<u>Primary Key</u>	<u>Foreign Key</u>	<u>Referenced Table/Column</u>	<u>Auto Incrementation</u>	<u>Default Value</u>	<u>Unique</u>	<u>Unsigned</u>	
StudentID	int(10)		YES	YES	Student(StudentID)			NULL	YES	YES
ProjectID	int(10)		YES	YES	DevelopmentProject(ProjectID)			NULL	YES	YES



