Leandro Lopez
Database Principles and Design
Merrimack College

Project 3

```sql
USE MealPlanning;
/* 1 */

DELIMITER $$

DROP FUNCTION IF EXISTS IngredientExists;$$
CREATE FUNCTION IngredientExists(ingredient_name VARCHAR(100))
RETURNS BOOLEAN DETERMINISTIC
BEGIN
    DECLARE cnt INT;
    SELECT COUNT(*)
    INTO cnt
    FROM INGREDIENTS
    WHERE IngredientName = ingredient_name;

    RETURN cnt > 0;
END $$

DELIMITER ;

/* 2 */
/* a */
CALL InsertNewRecipe("Empanadas", "Mom's Recipes", 10, TRUE, NULL)

/* b */
DELIMITER $$
DROP PROCEDURE IF EXISTS InsertNewRecipe;
CREATE PROCEDURE InsertNewRecipe (
    myRecipeName VARCHAR(100),
    myCookBookName VARCHAR(200),
    myTotalServings INT,
    myIsBook BOOL,
    myWebsite VARCHAR(200),
    myIngredients TEXT
)
BEGIN
    DECLARE ingredientName VARCHAR(100);
    DECLARE endOfIngredients INT DEFAULT 0;
    DECLARE ingredientCursor CURSOR FOR
        SELECT SUBSTRING_INDEX(SUBSTRING_INDEX(myIngredients, ',', numbers.n), ',', -1)
        FROM (
            SELECT 1 n UNION ALL SELECT 2
            UNION ALL SELECT 3 UNION ALL SELECT 4
            UNION ALL SELECT 5 UNION ALL SELECT 6
```

```sql
            UNION ALL SELECT 7 UNION ALL SELECT 8
            UNION ALL SELECT 9 UNION ALL SELECT 10
        ) numbers
        WHERE numbers.n <= CHAR_LENGTH(myIngredients) -
CHAR_LENGTH(REPLACE(myIngredients, ',', '')) + 1;
    DECLARE CONTINUE HANDLER FOR NOT FOUND SET endOfIngredients = 1;

    IF (SELECT doesCookbookExist(myCookBookName) = FALSE) THEN
        INSERT INTO Cookbook (CookbookName, IsBook, Website)
        VALUES (myCookBookName, myIsBook, myWebsite);
    END IF;

    IF (SELECT doesRecipeExist(myRecipeName) = FALSE) THEN
        INSERT INTO Recipe (RecipeName, CookbookName, TotalServings)
        VALUES (myRecipeName, myCookBookName, myTotalServings);
    END IF;

    OPEN ingredientCursor;

    ingredientLoop: LOOP
        FETCH ingredientCursor INTO ingredientName;
        IF endOfIngredients THEN
            LEAVE ingredientLoop;
        END IF;


        IF NOT IngredientExists(ingredientName) THEN
            INSERT INTO Ingredients (IngredientName) VALUES (ingredientName);
        END IF;


        INSERT INTO Meal (RecipeName, IngredientId)
        SELECT myRecipeName, Id FROM Ingredients WHERE IngredientName = ingredientName;
    END LOOP;

    CLOSE ingredientCursor;
END$$
DELIMITER ;

/* 3 */
DELIMITER $$
DROP PROCEDURE IF EXISTS recipeIngredients;$$
CREATE PROCEDURE recipeIngredients (
        myRecipeName VARCHAR(100)
)
BEGIN
        SELECT I.IngredientName
    From Ingredients I
    INNER JOIN Meal M ON I.Id = M.IngredientId
```

```sql
    WHERE M.RecipeName = myRecipeName
    ORDER BY I.IngredientName ASC;
END$$
DELIMITER ;

/* 4 */
DROP VIEW IF EXISTS Recipes_Cookbooks_Ingredients;
CREATE VIEW Recipes_Cookbooks_Ingredients AS
SELECT r.RecipeName, r.CookbookName, i.IngredientName
FROM Recipe r
JOIN Meal m ON r.RecipeName = m.RecipeName
JOIN Ingredients i ON m.IngredientId = i.Id
ORDER BY r.CookbookName, r.RecipeName, i.IngredientName;

/* 5 */
SELECT r.RecipeName, COUNT(i.Id) AS NumberOfIngredients
FROM Recipe r
JOIN Meal m ON r.RecipeName = m.RecipeName
JOIN Ingredients i ON m.IngredientId = i.Id
GROUP BY r.RecipeName
ORDER BY COUNT(i.Id) DESC;

/* 6 */
SELECT i.IngredientName
FROM Ingredients i
LEFT JOIN Meal m ON i.Id = m.IngredientId
LEFT JOIN Recipe r ON m.RecipeName = r.RecipeName AND r.RecipeName = 'MyFavoriteRecipe'
WHERE r.RecipeName IS NULL
ORDER BY i.IngredientName ASC;
```