**Activity 1**

I made three new tables: Campaign, Company and CampagnRecipe

The Company is just for storing companies that are connected to the database.

Campaign store data about the campaign name, what discount it should have, start and end date if needed, the companyID who creates the campagin and also if there should be a campaign just for a area, I added a start postalcode and a end postalcode. For example the campaign just are relevant for users living in a special area, they can add a postal code.

CampaignRecipe is just for redundancy so a recipe can be used in many campagins. Since I can get all the ingredients for a recipe from other commands.

**Activity 2**

CREATE PROCEDURE add\_campaign

 (@name varchar(255), @discount decimal(12,8), @startDate date, @endDate date, @companyID int, @startPostalCode int, @endPostalCode int, @recipeID int)

AS

BEGIN

 INSERT INTO Campaign

 VALUES(@name, @discount, @startDate, @endDate, @companyID, @startPostalCode, @endPostalCode)

 INSERT INTO CampaignRecipe

 VALUES

 ((SELECT MAX(campID) FROM Campaign)

 ,@recipeID);

END

GO

The part that was tricky is how to select the latest added campaign into the list. But after some thinking it was pretty obv that you can just take the maximum value from Campaign since its unique number and increments with 1 all the time.

**Activity 4**

**NOT NULL –** If a row is allowed to be null or not.

**UNIQUE** – If the value should be able to be unique

**PRIMARY KEY** - designated to uniquely identify all table records.

**FOREIGN KEY** – attribute that matches a candidate key of a relationship. Uses for referate to a another relation.

**CHECK** - The CHECK constraint is used to limit the value range that can be placed in a column.

ALTER TABLE Campaign ADD CHECK (startDate<=endDate);

In my database a CHECK is used too see if the start date is before the end date. So we dont create errors.

**DEFAULT** - The DEFAULT constraint is used to provide a default value for a column.

ALTER TABLE Menureview ADD CONSTRAINT CurrentDate DEFAULT (getdate()) FOR timestamp;

Default is used when a user adds a review on menu or recipe. So the date is when they enterd the recipe.

**CREATE INDEX** - Creates a relational index on a table or view. A special lookup table for database search engine to look up data quicker.

I could use it for recipe to store recipes after name in alphabetical order. If the database is large enough.

**STORED PROCEDURE** – Is a prepared SQL code that you can easily access. Easy too use for people not knowing SQL.

I use it in our database for adding employees, gets ingredients in a recipe etc.

**CASCADE UPDATE** – Updates all ”children” to the parent that being updated.

Cascade update should be used on primarykeys where you use text. Dont find it nesecsary to use it on Ids with integers.

**CASCADE DELETE**– DELETE all ”children” to the parent that being deleted.

In this database I dont see the use of cascade delete, since for example) I delete an employee, the recipe should still be there. I thought about having it in campaigns, but since campagins can be the same maybe 1 time a year, I still want it left.