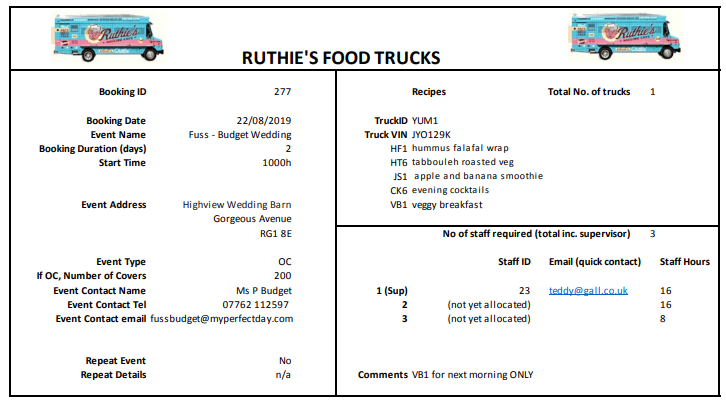
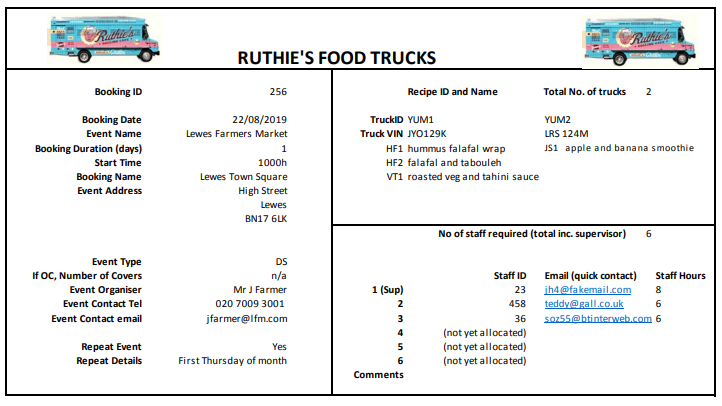
You will be supplied with a partial ERD after the feedback date for part 1, in case you choose to revisit your design before full implementation. Write a report which documents the implementation of the database to support the Information System that you designed in part 1. Your report must contain a short description of each stage of the implementation and explain the logic of the any procedural code that has been written. You must also provide the code itself and screen-shots to demonstrate that each part of your design works correctly. **You may be required to demonstrate these and will need to keep a copy of your implemented database until marks have been returned.**

# Appendix B:

## Ruthie’s food truck booking forms for normalisation:





**USE APPENDIX B**

## Description of task:

Write a report which documents the logical design of a database to support the Information System requirements described in Appendix A. Your logical design should be carried out using the top-down and bottom-up techniques that you have been taught in the module.

## Evidence of normalization:

Normalise the supplied booking document (removing repeating groups, resolving functional dependencies). See Case Study. Two instances of the booking form are supplied. The outcome of the normalisation should inform your ERD.

## 

## Definition of tables / Data dictionary:

For each table, list the table name, a description of the entity that is implemented, the primary key of the table (name and data type) and other identified attributes (names, data types and all other constraints).

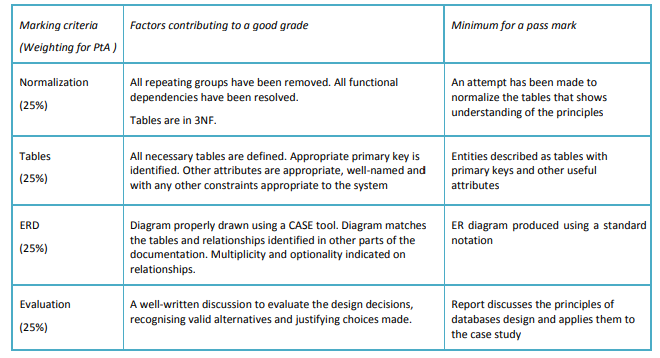
## Entity-Relationship Diagram to complete the logical design:

Produce a key-only Entity-Relationship diagram showing primary and foreign keys to meet the requirements, showing all entities and relationships (with relationship name and cardinality).

## Evaluate your design:

Write a short (250 word) evaluation of your database design, discussing the requirements that you have included and any problematic issues that were difficult to resolve.

50% coursework mark:



# Implementation:

**USE APPENDIX A**

## Create tables using SQL DDL:

Write SQL statements to create the tables that implement the database you designed in part 1. Your report must show the SQL CREATE TABLE statements and images of the tables created in SQL Server or mySQL or another DBMS approved by the module tutor.

## Populate tables using SQL insert:

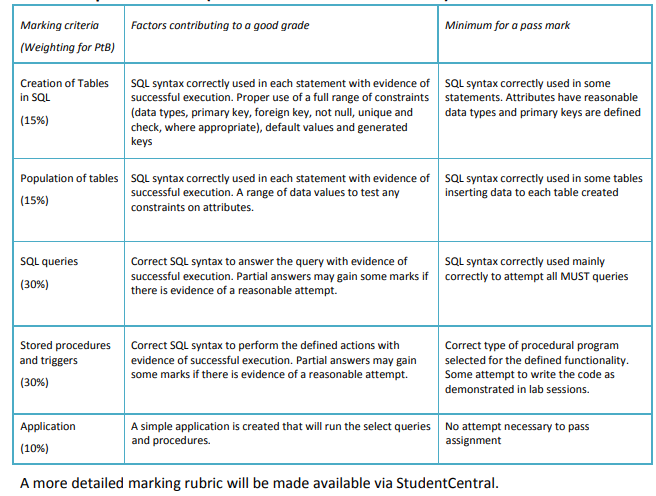
Write SQL statements to populate all the tables you have previously created, with at least five records each. Your report must show the SQL INSERT statements and images of the tables when they have been populated. NOTE: you should consider creating a formal test data set where you know what the outcome of a given SQL query will be.

## Retrieve information using SQL queries:

Write SQL queries to demonstrate that your system can address the requirements in Appendix A. Your report must show the SQL code and the result of executing each query. These should be clearly labelled with the requirement number.

## Implementation of DMBS functionality:

Design appropriate stored procedures, functions, triggers and application to implement the functionality defined in Appendix A. Your report must show the SQL code and the result of execution.



# Appendix A: Requirements for the database system

Initial consultation has resulted in the following prioritised requirements. The system:

### MUST:

1. Record /edit/cancel bookings for multiple trucks at outdoor events
2. For all staff, ensure that a level 2 food hygiene certificate is in place
3. For each item sold, produce a list of ingredients
4. Identify all truck bookings for any given date
5. Identify all bookings for a given truck

### SHOULD:

1. Record any accidents / reportable incidents that take place, including the event at which they happened
2. For all staff acting as supervisors, ensure that a level 3 food hygiene certificate is in place
3. For all events involving alcohol, identify a team member as the license holder
4. Identify any temporary staff who are not currently working this weekend (i.e. may be available for a last minute booking)

### COULD

1. Record members of staff who worked at specific events, including who was acting as the supervisor. This should include the number of hours worked by each staff member, and their rate of pay
2. Produce a list of ingredients for each item sold, with known allergens at the top of the list - this should include sub-ingredients, like the tahini ingredients on the hummus recipe described previously

### WOULD BE NICE IF:

1. Identify weekends in the next month with available trucks (to offer last minute deals)
2. For hourly paid staff, calculate wages due for the hours worked in the past month [wages = (hours worked \* hourly rate) – tax] - you may assume that 20% tax is paid directly to HMRC and should subtract this amount from the Gross Wages and record it, and be able to calculate Net Wages to be paid.

The following functionality is also required in the database system:

* 1. a stored procedure is required that will create a new booking using parameters giving the details of the date and length of event in days, a named contact and the total number of workers required
  2. stored procedure is required to produce a list of bookings and staff members (identifying the supervisor) for the forthcoming month
  3. a trigger is required that is actioned when a booking is cancelled. This will produce all staff who are booked to be working at this event, with their contact details so that that they can be alerted to change
  4. Create a simple application to provide an interface capable of executing the above functionality (select queries and procedures)

\*For more information on the 14 allergens that must be listed, see here:

<http://allergytraining.food.gov.uk/english/rules-and-legislation/>

\*\* For more information on RIDDOR, see here:

<https://www.thecaterer.com/articles/45/accident-reporting-in-thehospitality-industry>

# Questions to ask before submission date: