



Assignment 2 - Database Implementation - Monash Food

Monash Food operates a restaurant on the Clayton Campus overlooking the Monash Common. They have asked that their current paper based recording system be computerised.

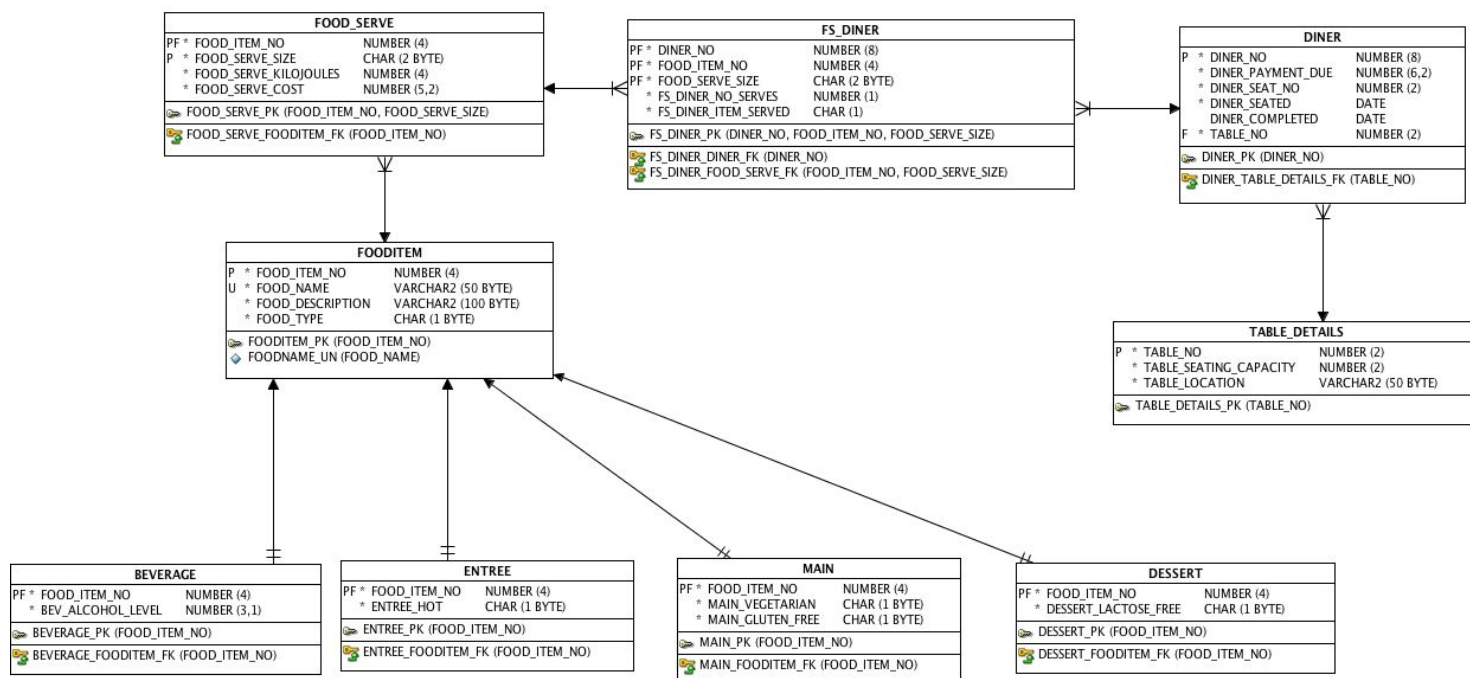
The restaurant has a number of tables for which they would like to record the table number, its seating capacity (how many diners/people can sit at the table) and the table's location within the restaurant.

When a diner (person) arrives at the restaurant they are escorted to their table and assigned a seat number at the table to help waiting staff keep track of the meals to be served. Each table's seat numbers start at one and increase by one until they reach the table's seating capacity. Monash Food is not interested in storing any personal details about a diner, nor in tracking a diner to see if they return to the restaurant (each visit is treated as a new diner). A table may have several different groups seated at it across the evening as one set of diners finish their meal, leave the restaurant and a new set of diners arrive. Monash Food needs to track what *each* diner orders and how much they owe for what they have ordered.

Monash Food offer a wide range of food items. For each food item they wish to record the name of the item, a description and the cost per standard serve. If the food item is a Beverage they wish to record the alcohol level. If the food item is an Entrée they wish to record whether the item is a hot or cold entrée. If the food item is a Main they wish to note if the item is vegetarian and also if it is gluten free. As well as the standard serve size some mains are also available as a small serve, some also as a large serve and some are available as standard, small and large serves. Monash Food wishes to record the kilojoules and serve cost for the small and large mains. If the food item is a Dessert they wish to record if the item is lactose free. For all beverages, entrées, mains and desserts they wish to record the kilojoule value of the standard item.

Diners order food items, which are prepared by the kitchen staff and then delivered to the diners at their table. Some diners who might be especially hungry or big eaters may order more than one serve of a particular food item (eg. two serves of chocolate mousse).

A data model has been developed for this situation. The relational model is shown below:



A full sized copy of the model is available as a PDF document from Moodle.

When a new diner is seated at the restaurant an entry is made into the DINER table. The diner_seated attribute of DINER is set and they are assigned a table and seat no at that table, the diner_completed is not known since they have not completed their meal. A diner orders items from the menu which are recorded in the table FS_DINER. Within this table fs_diner_no_serves indicates the number of serves they have ordered and fs_diner_item_served is set to 'O' to indicate this is an order. When an item is served to a diner their payment due is updated and fs_diner_item_served is set to 'S' to indicate the order has been served. After the diner has completed their meal they pay their payment due and the diner_completed attribute is set to signify the close of this diner experience.

To simplify this task we are applying a rule that says diners can only order any particular item of food only *once* - however they may order multiple serves when they place this order.

You have been supplied with a schema file MonashFood-schema-start.sql (which **must not be altered in any way**) which partially implements the Monash Food model.

You have also been supplied with a document ass2-solution.sql - you should rename this script by prepending your authcate username to the start of the filename eg. abc123-ass2-solutions.sql. This script file will be referred to as your *solutions script*. Within this script there are marked points where each of your solutions must be added.

All of the work for assignment 2 will take place in this document so ***please take great care to keep regular backups, including off your computer eg. on Google Drive, so you do not lose work.***

Before starting work on the task complete the header by adding your name etc, in the solutions script.

In completing this assignment you are not permitted to manually look up a value in the database, obtain its primary key (for example) and use that in your answer. As an example you cannot look in the database and see that 'Bruschetta' is food_item_no 1 and use this in your work. ***You must use only the values listed in this document for the particular task you are working on.***

TASK 1: Data Definition [20 + 5 = 25 mks]

For this task you are required to complete the following:

- 1.1 Add to your solutions script the CREATE TABLE and CONSTRAINT definitions which are missing from the MonashFood-schema-start.sql script. You **MUST** use the entity and attribute names shown in the data model above to name tables and attributes which you add.
- 1.2 Add the full set of DROP TABLE statements to your solutions script. In completing this section you may only use DROP TABLE tablename - you are not permitted to add any other clauses such as cascade constraints

Before proceeding with Task 2 you must run the supplied MonashFood-schema-start.sql (which **must not be altered in any way**) followed by the extra definitions that you added in 1.1 above.

In a script you can run a section of the script by highlighting the lines you wish to run and selecting the run button. If at any stage your tables are corrupted during working on this assignment you simply need to run your drop commands from 1.2 above and then rerun MonashFood-schema-start.sql and your extra definitions that you added in 1.1 above.

TASK 2: Data Manipulation [12 + 8 + 5 = 25 mks]

Run the script MonashFood-insert.sql to add some initial data into the tables you created in task 1.

Data manipulation tasks:

- 2.1 Add to your database four DINER records and their associated FS_DINER records.

These four diners should all represent *completed* dining experiences (ie. they have ordered, been served, paid the full amount outstanding and left the restaurant) which occurred in May 2017. You may pick any range of dates/times you wish for these diner records.

Each diner must have a minimum of two FS_DINER records, you may pick any food items and number of serves you wish, however ensure you use a variety of items and serves.

For diner_no's you may assign primary keys that you choose provided the number is *below* 10 (ie. the values must be in the range from 1 to 9). All four diners should be assigned to table number 1, over a range of dates. Table 1 has a seating capacity of 3 (assign an appropriate seat number from 1 .. 3). For **this question only**, you may use the following food item details:

FOOD_NAME	FOOD_SERVE_SIZE	FOOD_SERVE_COST
Arancini Balls	ST	12
Beef Carpaccio	ST	12
Braised Beef Brisket	LG	45
Braised Beef Brisket	SM	28
Braised Beef Brisket	ST	38
Bruschetta	ST	12
Classic Chocolate Mousse	ST	10
Corona Beer	ST	12
House White (Glass)	ST	12
Ravioli Ricotta	SM	18
Ravioli Ricotta	ST	28
Spaghetti Marinara	ST	33
Sticky Date Pudding	ST	14

- 2.2 An Oracle sequence is to be implemented in the database for the subsequent insertion of records into the database for the tables FOOD_ITEM and DINER. Provide the CREATE SEQUENCE statements for the FOOD_ITEM and DINER tables. The sequences will be used to generate new primary key values when adding new tuples/rows to the database:
- The sequence for FOOD_ITEM should start at 11 and increment by 1
 - The sequence for DINER should start at 10 and increment by 1
- 2.3 Provide the DROP SEQUENCE statements for the sequence objects you have created in Q2.2 above

TASK 3: Database Insert/Updates [5 + 5 + (5 + 5 + 5) = 25 mks]

Sequences created in task 2 must be used to insert data into the database for the task 3 questions. For these questions you may only use the data supplied in this task.

- 3.1 Add food: Add a new DESSERT to the Monash food menu - you will need to research some meaningful data to be able to add this item. DESSERT's are food_type 'D' and are only served in standard 'ST' serve sizes.
- 3.2 Increase Price: Monash food has decided to increase the price charged for all standard serve ('ST') main food items ('M' food type) by 15%, make this change in the database
- 3.3 Diner activity:
 - a. A new diner has just arrived and been seated at Table 1 seat 3. Update the database to seat this diner.
 - b. This new diner calls the waiter over and proceeds to order two 'Bruschetta' entrees. Entrees are only available in a standard 'ST' size. Add this data to the Monash Food System for this diner. The food item has not been served as yet, this is an order only
 - c. Some time after this order has been recorded the 'Bruschetta' ordered in (b) is served to this diner - update the database to record this service.

TASK 4: Database Structure [10 + 15 = 25 mks]

- 4.1 After using the system for some time Monash Food has realised that it is not making the optimal use of its diner information and so has decided that it will offer all future diners an opportunity to provide their name, contact mobile number and email address so that Monash Food can let them know about specials and events which they run (diners are not required to provide this information). Change the "live" database so that this information can be recorded. For any structures you create you are not required to include column comments.
- 4.2 After the start of *each* new financial year (July 01) Monash Food wish to archive diner (DINER) and order (FS_DINER) information into two historical data tables and remove all the current data in these two tables representing the previous financial year. This archive will be carried out some time early in the new financial year, all completed diner records for the past financial year must be archived.

This change should be carried out on your database after 4.1 has been successfully completed.

DINER should be archived into DINER_HISTORY and FS_DINER into FS_DINER_HISTORY. For the diner archive (DINER_HISTORY) the seat number and table number attributes will *not* be stored in the history table. For the FS_DINER table all attributes must be archived into the history table. The relationship between the two history tables must be maintained. The history tables may be created without column comments.

ALL foreign key relationships between these history tables and any already existing tables in the database must also be maintained:

- FS_DINER_HISTORY must maintain a relationship to FOOD_SERVE, and
- DINER_HISTORY must maintain a relationship to the table/s you created in 4.1 above.

Submission Requirements

Due: Friday 6-OCT-2017 11 PM (week 10)

For this assignment there is only one file to submit. You are required to submit your solutions script file to Moodle before the assignment due date/time. If you need to make any comments your marker/tutor should be aware of please place them at the head of your solutions script in the "Comments for your marker:" section.

Late submission will incur penalties as outlined in the unit guide.