

## Chelsea Pizza Database



# Investigate

## Project 1 Development Schedule

Project 1 Development Schedule																																																				
Andrew Dreyer																																																				
Month																																																				
Week																																																				
Date	24	25	26	27	28	29	30	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27																		
	11				Holidays											1					2					3					4					5					Mid Term Break					6					7	
Investigate																																																				
Write a brief outline of the problem																																																				
Written description																																																				
Ethical Legal & Security Issues																																																				
Factors That Impact the Quality of Stored Data																																																				
Programming and Database Requirements																																																				
Must have requirements																																																				
Should Have Requirements																																																				
Nice to have Requirements																																																				
Design																																																				
Design the Entity-Relationship Diagram																																																				
Create the Data Dictionary																																																				
Create Plain English Queries																																																				
Decribe Validation Rules																																																				
Convert Word Doc to PDF																																																				
Develop																																																				
Script to Create an Empty Database																																																				
Script to insert sample data for testing																																																				
Add Data Validation to Ensure Integrity																																																				
Create Website to Interact with the Database																																																				
Create SQL Queries that meet all requirements																																																				
fix structure and use good programming practise																																																				
Evaluate																																																				
Consider how well my solution met the requirements																																																				
Discuss The Structure and Any Changes of the Database																																																				
Consider how I could improve my user experience																																																				
Document bugs & limitations and explain their impact																																																				
Reflect on if I followed my plan and if it Was Realistic																																																				
Reflect on If I did Enough Preparation																																																				
Reflect If my Design Was Adequate During Development																																																				
Proofread & clean for submission and convert to PDF																																																				

## Project Outline

The purpose of this project is to design a relational database system for Chelsea Pizza. Chelsea Pizza is a Pizza restaurant located at 145 Stirling Hwy, Nedlands WA 6009. Chelsea Pizza is actually 2 separate businesses, Chelsea Pizza (<https://chelseapizza.com.au>) and Grissini's restaurant (<https://grissiniperth.com.au>). Grissini's restaurant is the walk in part of the restaurant and is open on Tuesday – Sunday from 5-10pm, and Chelsea Pizza is open from 5-10pm every day, except 12am-10pm on Friday. Chelsea Pizza offers takeaway pizza and supplies pizzas to the restaurant for walk in customers.

For this project, I will treat both businesses as single business, and design a single database for both businesses. Grissini's restaurant also has a wider menu on top of the pizza menu, but I will treat them as one menu, but only offer the menu items on the times that they are open. I will create a database as well as a website to interact with the database for customers to order and employees to view shifts, and take orders for customers.

## Problem Description

For this project, I will have many considerations. For example, there are many factors that affect the price of an order. When a customer is ordering a pizza, they should be able to add or remove certain toppings from the pizza, and adding different toppings can incur extra costs depending on what ingredients are added. The customer is also able to choose between a medium or large pizza, as well as a gluten free base (only offered for medium sized pizzas), which all have different prices. Certain items also have delivery costs, and the minimum price to spent to qualify for delivery is \$40. When making the webpage, I will have to create a separate page for customers and employees, where the customer is able to place orders and view the menu, but the employees are also able to view their shifts and will also need to put in orders on behalf of customers, when taking orders over the phone.

## Issues:

### Ethical

In my database, I will ensure that I am using the data as ethically as possible. Here are the considerations I am making with the data:

1. **Transparency.** I will let customers know what data is being stored about them by having a disclaimer at the bottom of the website that will tell them what data is being stored about them, and who has access to that data.
2. **Data Minimisation.** I will only collect the minimum amount of data needed to get customers their food. If the customer is dining in, I will not collect any personal information and only record what they have bought and how much they have spent at the restaurant. If they make a reservation or get their food delivered, I will only record the minimum information to make sure they get their reservation or get their food delivered.
3. **Anonymisation and De-identification.** I will make sure that when recording purchase histories, I only store what was purchased and how much was spent, and not tie the data to any customers at all, in order to keep customer information private.
4. **Privacy by Design.** I will design my database with privacy in mind, by only allowing the customer data to be shown to employees that need to see the data. For example, only the delivery driver will be able to see a customers address in order to deliver their food to the customer.

5. **Consent For Data Sharing.** If the data is ever shared to a third party, I will make sure to obtain consent from all users involved and only share the minimum amount of data needed for that purpose.

## Legal

When making my database, I will have to follow the Privacy act of 1988, including the 13 Australian Privacy Principles (APPs). Here are the key implications of the act that I have considered:

1. **Data collection and consent.** If a customer is dining in, my database will not require them to have a customer ID, and will not record any of their personal information, and will only record the food they order and the invoice receipt. Otherwise, the only information that my database collects about customers is their first and last name, their phone, and their address. I will only ask for a customers address if they are having their food delivered, as this is required to get them their food, and they will be aware that I am storing their address, in order to deliver their food. Their first and last name will be recorded in order to know who has made each reservation or know that the right person is getting their food delivered. Their phone number will also only be required if they are having their food delivered, in order to notify them that their food has arrived.
2. **Data Breach Notification.** If my database is breached in a way that is likely to result in serious harm towards the individuals involved, I will take steps to ensure the customers are aware of this breach, by using the stored phone number or address in order to alert them of the breach, as well as notifying the Office of the Australian Information Commissioner as soon as possible.
3. **Data Accuracy and Access.** I will aim to make sure the database is up to date by updating customer information whenever they make a new order and allow customers to see the data we are storing about them if requested and allowing them to request corrections with the data we are storing about them if necessary.

## Security

Because the database contains personal information about customers and employees, as well as other confidential information of the business, it is important to ensure the database is properly secured. The steps that I will take to keep the data secure include:

1. **Data encryption.** I will keep all information in the database encrypted so even if the data is compromised the attackers are not able to access the data properly.
2. **Access Controls.** I will have strict access controls and authentication measures to ensure that only authorised personnel are able to modify the data of customers. I will do this by using the Principle of Least Privilege, which only allows users to have the minimum required access to the database they need to perform their tasks.
3. **Auditing and Monitoring.** I will regularly monitor the database for unusual activity in order to detect and respond to potential breaches that may appear in my database system.
4. **Backup Storage.** I will have a backup storage of all data recorded in my database that is also encrypted and regularly tested. This will prevent the theft of data from severely impacting the business by allowing the data to be quickly restored.
5. **Input Sanitisation and Validation.** I will sanitise all input that comes into my database and use parameterised queries to ensure that my system is less vulnerable to SQL injection attacks.

## Factors Affecting the Quality of Stored Data

One of the main factors that could affect the data quality is duplicate data, as it could lead to update, delete, or insert anomalies. I have mitigated the risk of this affecting my data by aiming to get my data into third normal form, which makes duplicate data less of a problem. Another factor affecting data quality is its accuracy. This could happen when an employee accidentally takes an order wrong, or a customer lies about their name. I can mitigate this through the use of constraints on my database, which reduces the risk of data being incorrectly inputted. I also have to consider the relevance of the data to my database, and I have removed all fields that are not relevant to the business or database. Although I am not able to trust the authenticity of customers inputting their name into my database, I don't need to confirm that is their name because their real name is not required for them to make a booking or have their pizza delivered, but I should also allow customers to be able to change their name that is stored in the database if necessary. The last factor I will need to consider is the currency of my data. Some of the fields may be changing regularly, such as the price that the business is paying for ingredients or the delivery fee, so there will need to be a system in place to make sure that when a price or other field is updated on the website or manually, the price stored in the database is also changed.

## Programming Requirements

### Must Have

- A website that allows the user to view the menu.
- A webpage for employees to see the menu.
- A way to let the users order from the menu.
- Input sanitisation of all inputs from customers on the website.
- Letting the users see the total they are spending for their order.
- A python script to create an empty database.

### Should Have

- A way to let users see what specials are currently available.
- A way for employees to find out when they have their shifts.
- Create a script to insert sample data into the table.

### Nice to Have

- A calendar for employees to see shifts and put in their availability.
- A way to view past orders that have been made by customers.
- More CSS to make the website more user friendly.

## Database Requirements

### Must Have

- A table for customer and employee information
- A table for the items that have been ordered.
- A table with all of the items on a menu.

### **Should Have**

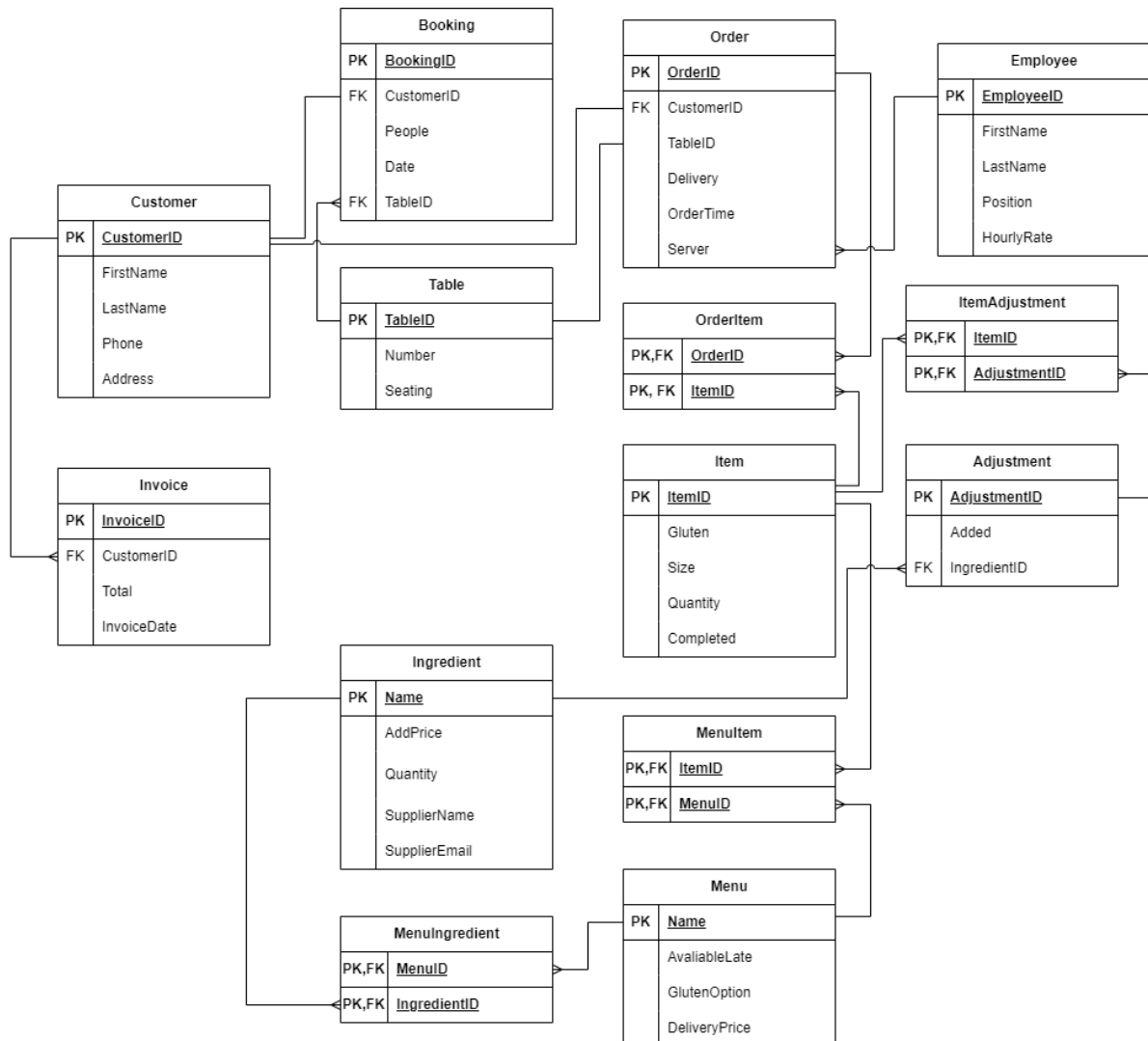
- A table of the adjustments that a customer wants to make to their order.
  - A table for reservations that have been made.
- A table for all of the ingredients, so the inventory can be properly managed.

### **Nice to Have**

- A table with the active specials as records. This may be hard to implement if I want to have a temporary menu item.
  - A table recording all invoices that customers have made.
  - A table to store all of the shifts that employees have.

# Design

## Entity-Relationship Diagram



## Data Dictionary

I have thirteen tables in total, but I have used four of the tables to resolve the many to many relationships between my tables, and so my data dictionary will have a brief overview of the other nine tables. I am not including the size of each of my datatypes because MySQL converts all datatypes to one of five different types (see <https://www.sqlite.org/datatype3.html>), so there is no point adding sizes because MySQL does not impose any length restrictions.

## Menu:

the menu table stores information about each item available on the menu, including its name, when it is available and other key information.

Field	Description	Data Type	Constraints	Key
Name	The name of the item on the menu	VARCHAR	Unique, Not Null	PK
AvailableLate	A few menu items are only available on Tuesday to Friday until 9pm. True if it is available late	BOOL	None	
GlutenOption	If a gluten free option of the meal is available	BOOL	None	
DeliveryPrice	The added price it costs to get that menu item delivered	FLOAT	None	
BasePrice	The base price of the menu item, which increases when ingredients are added to the pizza, the size is changed, or a gluten option is selected	FLOAT	None	

## Ingredient:

Each individual ingredient that is in a menu item. Details who supplies the ingredient, the current quantity of ingredients available, and the cost to add the ingredient to the pizza.

Field	Description	Data Type	Constraints	Key
Name	The name of the ingredient	VARCHAR	Unique, Not Null	PK
AddPrice	The cost to add that ingredient to a pizza.	FLOAT	None	
Quantity	The number of portions available in the inventory now	SMALLINT	Not Null	
SupplierName	The name of the company supplying the ingredient	VARCHAR	None	
SupplierEmail	The email of the company supplying the ingredients	VARCHAR	None	

## Item

Each record is a menu item that is part of an order, and contains specifications about that item, including its size, the amount ordered, and if they chose a gluten-free option.

Field	Description	Data Type	Constraints	Key
ItemID	A unique identifier for each menu item that has been ordered	SMALLINT	UNIQUE, NOT NULL	PK
Gluten	If the item is meant to be gluten free	BOOL	None	
Size	The size selected, either medium or large	CHAR	None	
Quantity	The number of times this item has been ordered	SMALLINT	None	
Completed	If the item has been cooked or not	BOOL	None	



## Adjustment

Specifies the adjustments in ingredients that a customer has made to each item in their order, shows what ingredients changed from a menu item.

Field	Description	Data Type	Constraints	Key
AdjustmentID	A unique identifier for each adjustment made	TINYINT	UNIQUE, NOT NULL	PK
Added	True if the ingredient is added to the item, False if it is removed.	BOOL	NOT NULL	
IngredientID	The ingredient that is being added or removed	SMALLINT	NOT NULL	FK
ItemID	The item the adjustments are associated with	SMALLINT	NOT NULL	FK

## Order

The order table is the orders made by customers. It specifies who made the order, if the order is being delivered, and there is only one order per table or delivery. The server ID is required if the customer is eating at the restaurant, which I will implement through my programming.

Field	Description	Data Type	Constraints	Key
OrderID	A unique identifier for each order made.	SMALLINT	UNIQUE, NOT NULL	PK
CustomerID	The customer that has made the order	MEDIUMINT	None	FK
TableID	The table the order is associated with. Null if the order is delivered or picked up.	TINYINT	None	FK
Delivery	True if the order is being delivered, False if they are eating in.	BOOL	NOT NULL	
OrderTime	What time the order was taken at.	DATETIME	None	
Server	The employee that took the order.	MEDIUMINT	None	FK

## Employee

This table is for all the employees, and records their name, job role, and hourly rate.

Field	Description	Data Type	Constraints	Key
EmployeeID	A unique identifier for each employee	MEDIUMINT	UNIQUE, NOT NULL	PK
FirstName	The first name of the employee	VARCHAR	NOT NULL	
LastName	The last name of the employee	VARCHAR	NOT NULL	
Position	What position in the company the employee has.	VARCHAR	NOT NULL	
HourlyRate	How much the employee is paid per hour	FLOAT	None	

## Customer

Information about the customers that is used for taking reservations or making deliveries.

Field	Description	Data Type	Constraints	Key
CustomerID	A unique identifier for each customer	MEDIUMINT	UNIQUE, NOT NULL	PK
FirstName	The first name of the customer	VARCHAR	NOT NULL	
LastName	The last name of the customer	VARCHAR	NOT NULL	
Phone	The phone number of the customer	VARCHAR	None	
Address	The address of the customer for delivery	VARCHAR	None	

## Invoice

A history of the purchases that have been made in the restaurant, which can be used to find the revenue of the business over time.

Field	Description	Data Type	Constraints	Key
InvoiceID	A unique identifier for each invoice	INT	UNIQUE, NOT NULL	PK
CustomerID	The customer that paid for the invoice	MEDIUMINT	NOT NULL	FK
Total	The total the customer paid in the order	FLOAT	NOT NULL	
Invoice Date	The date the order was paid for	DATETIME	None	

## Table

Information about each table in the restaurant.

Field	Description	Data Type	Constraints	Key
TableID	A unique identifier for each table	TINYINT	UNIQUE, NOT NULL	PK
Number	What number the table is (may change from night to night)	TINYINT	NOT NULL	
Seating	How many the table seats	TINYINT	NOT NULL	

## Booking

Field	Description	Data Type	Constraints	Key
BookingID	A unique identifier for each booking made	INT	UNIQUE, NOT NULL	PK
CustomerID	The customer that made the order	MEDIUMINT	NOT NULL	FK
People	How many people are booked in	TINYINT	NOT NULL	
Date	When the order is booked for	DATETIME	NOT NULL	
TableID	The table the booking is made for	TINYINT	NOT NULL	FK

## Relational Notation

**Menu**(Name PK, AvailableLate, GlutenOption, DeliveryPrice)

**Ingredient**(Name PK, AddPrice, Quantity, SupplierName, SupplierEmail)

**Item**(ItemID PK, Gluten, Size, Quantity, Completed)

**Adjustment**(AdjustmentID PK, Added, IngredientID FK)

**Order**(OrderID PK, CustomerID FK, TableID FK, Delivery, OrderTime, Server)

**Employee**(EmployeeID PK, FirstName, LastName, Position, HourlyRate)

**Customer**(CustomerID PK, FirstName, LastName, Phone, Email)

**Invoice**(InvoiceID PK, CustomerID FK, Total, InvoiceDate)

**Table**(TableID PK, Number, Seating)

**Booking**(BookingID PK, CustomerID FK, People, Date, TableID FK)

**MenuIngredient**(MenuID PK FK, IngredientID PK FK)

**MenuItem**(MenuID PK FK, ItemID PK FK)

**ItemAdjustment**(ItemID PK FK, AdjustmentID PK FK)

## Plain English Queries

1. Select all items that need to be made for table five.
2. Find all employees that are working tonight, ordered by when they start working.
3. Find how many customers have made bookings from 6-7pm tomorrow night.
4. Change the customers reservation from 5pm to 8pm.
5. Delete all orders from the database that have had all items served.
6. Find the total of all the invoices dated between two weeks ago and one week ago.
7. Add all of table sevens orders to the database.
8. Find The average amount spent per person in the last month.
9. Increase Employee X's pay by \$2.00 an hour.
10. Find how many of each menu item has been ordered in the past year ordered by how many was ordered.

## Validation Rules

I have included the SQL constraints in my data dictionary, and so this section will only contain pseudocode with my programming validation rules.

```
1  FUNCTION ValidateItem(item)
2      cursor.execute("SELECT Name FROM Menu")
3      menuItems = cursor.fetchall()
4      IF item in menuItems:
5          processData(item)
```

```
6         END IF
7     END ValidateItem
```

```
1  FUNCTION ValidateAdjustment(ingredient)
2      cursor.execute("SELECT Name FROM Ingerdient")
3      ingredientList = cursor.fetchall()
4      IF ingredient in ingredientList:
5          processData(ingredient)
6      END IF
7  END ValidateAdjustment
```

```
1  FUNCTION ValidateName(FirstName, LastName)
2      validCharacters = "qwertyuiopasdfgklzxcvbnm -"
3      validName = True
4      FOR char in FirstName.lower():
5          IF char NOT in validCharacters:
6              validName = False
7          END IF
8      END FOR
9      FOR char in LastName.upper():
10         IF char NOT in validCharacters:
11             validName = False
12         END IF
13     END FOR
14     if validName == True:
15         ProcessData(FirstName, LastName)
16     END IF
17 END ValidateName
```

## **Appendix:**

### **Interview Questions**

- Q. When do the employees start and finish work?
  - a. employees start work at 4:30pm, half an hour before the restaurant opens, and finish at ~10pm, when the restaurant closes. I will have to consider this when creating the shift schedule.
- Q. How many employees do you have in total?
  - a. 27 Total. 3 managers, 5 people on the oven/ making the pizzas, 12 delivery drivers and 6 people on the phone (Names have been omitted for privacy reasons).
- Q. How many employees are working at one time, and what are their roles?
  - a. 3 Monday, 3 Tuesday, 4 Wednesday, 5 Thursday, 10 Friday, 7 Saturday and 5 on Sunday.
- Q. How many orders do you get per night?
  - a. Hundreds, but it depends on the day.
- Q. How many items do you get per order on average?
  - a. 2 – 3 is the most common for people to order.
- Q. When do the suppliers come and drop off the food?
  - a. They come every Tuesday afternoon at 1pm to drop off the food.