SQL Data Architecture Report: C-Lovers Fish & Chips

2023

APRIL 5, 2023

SFU Beedie – BUS 464 D200
Business Data Management
Professor: Aishwarya Deep Shukla
Authored by: Jaiden Angeles and Tianci Qiao

SFU
SIMON FRASER UNIVERSITY
ENGAGING THE WORLD

Table of Contents

Introduction

This project aims to construct a real-world MySQL database for C-Lovers Fish & Chips, a restaurant franchise native to British Columbia. Utilizing field research, online resources, and AI-assisted programming, we first created an ER-Diagram and capturing all relevant entities and attributes. We then inserted publicly available data and made reasonable assumptions where data was unavailable. A series of SQL queries were developed to carry out use cases and generate reports that a store manager might request. The project demonstrates the power of combining human insight and AI assistance in database architecture for real-world businesses.

Business Overview

C-Lovers Fish & Chips is a restaurant franchise native to British Columbia, having 11 different locations spread through the province. Since they are a franchise, their business operations through B.C. are almost identical, meaning observing each of these restaurants will lead to the same analysis and conclusion about their business processes and operations. However, the Coquitlam location was chosen to be observed because the city is their founding restaurant location. In the kitchen, they deep fry almost every restaurant item, not including their beverages or any sides (See Appendix 3 and 4 for their menu). Since their products are limited to fish and fries mostly, and their customization is low, observing their business processes and constructing use cases was made easier.

Field Study Information

Through visiting the restaurant to create our ERD and use case, we observed three types of interactions in the restaurant: Customer Interactions, Employee Interactions, and Supplier Interactions. Afterwards, we constructed an ERD with any other information we missed through online sources and C-Lovers Website (See Appendix 1 for their website).

Customer Interactions

The restaurant's target audience catered to a mature audience (typically 50 and above) due to the classic style of the restaurant, while using their "All-you-can-eat" special to attract a younger demographic. Analyzing their pricing options, they accept cash, credit, debit, and gift cards (which can be purchased or given away through a contest). Regarding their seating, they have 18 total tables, 6 of which seat four people while 12 of which seat two people, creating a capacity of 48 with no outdoor area available (See Appendix 2 for their location view). Customers can use coupons, which C-Lovers frequently uses in newspapers, to get discounts. Reservations can also be made. C-Lovers does not record any customer information in the restaurant, and payment information is not stored in any database.

Employee Interactions

The restaurant has seven positions within the store: prep worker, manager, assistant manager, fry cook, cashier, dishwasher, and waiter/waitress, with some working part-time shifts of around 3 hours while others worked full-time shifts at 8 to 9 hours. Employees are paid biweekly by cheque, have a tipping system, and label any days that they are unavailable so the manager can set the availability for the convenience of the restaurant. They are also mentored by a veteran on how to properly do the role (usually for 2 weeks) before going into the role on their own.

Supplier Interactions

Suppliers come on a weekly or biweekly schedule depending on the product, typically every Monday or Wednesday; some products are bought locally from a grocery store like milk. They store items three ways: via fridge, freezer, or cabinet. These include any supply needed for their menu items (see appendix 3 for their menu), which includes Salmon, Halibut, Haddock, Cod, Potatoes, Mushrooms, flour, and so forth.

Use Cases

In this report, we have outlined the major steps of each use case of C-Lovers for the sake of clarity and understanding. However, it is important to note that the actual implementation of these use cases is more comprehensive when using SQL queries. The numbers for each use case are classified “front-end” and “back-end” and match up with the use cases in the C-Lovers SQL Queries.

Uses cases:
Use case 2: Customer Interactions with menu and reservations (Front-end)
Use case 3: Customer Order management (Front-end)
Use case 4: customer checkout and refund management (Front-end)
Use case 1: Restaurant Inventory and Supplier Management System (Back-end)
Use case 5: Employee Shift Management (Back-end)

Front-end (i.e., customers, sales, and payments)

Use Case 2: Customer Interactions with Menu and Reservations

<table>
<tr><td colspan="2">Use Case Name: Customer Interactions with Menu and Reservations</td><td colspan="2">ID Number: UC-2</td></tr>
<tr><td colspan="4">Description: This table represents a comprehensive use case covering all the customer interactions with menu and reservations.</td></tr>
<tr><td colspan="4">Trigger: A customer request
Type: External</td></tr>
<tr><td colspan="2">Major Inputs:</td><td colspan="2">Major Outputs:</td></tr>
<tr><td>Description</td><td>Source</td><td>Description</td><td>Destination</td></tr>
<tr><td>customer name
phone number
reservation date
reservation time
party size
table id
reservation id
cancellation status</td><td>Customer Database
Customer Database
Restaurant Database
Restaurant Database
Restaurant Database
Customer Database
Restaurant Database
Restaurant Database</td><td>New reservation entry
Updated reservation entry with cancellation status
Menu item details
ingredients
alternative items
combinations
All You Can Eat menu items with special requests</td><td>Restaurant Database
Restaurant Database
Customer
Restaurant Database
Restaurant Database
Customer</td></tr>
</table>

cancellation date	Restaurant Database		

Major Steps Performed:	**Information for Steps:**
1. INSERT statement to create a new reservation. 2. UPDATE statement to cancel a reservation (if needed) 3. SELECT statements to retrieve menu items, ingredients, alternative items, combinations, and All You Can Eat menu items with special requests	Reservation information (IP) Reservation cancellation status (IP) Restaurant menu with all food information (OP)

Exceptions:

E1: The customer does not able to make reservation or reservation is full and they may like the food in menu (occurs at step 3). Therefore, the reservation will not be created, and use case stop.

Use Case 3: Customer Order Management

Use Case Name: Customer Order Management	**ID Number**: UC-3
Description: This table represents a comprehensive use case that covers all the functionalities related to customer order management, including adding a new order, order items, special requests, and linking special requests to order items.	
Trigger: A customer request **Type**: External	

Major Inputs:		**Major Outputs:**	
Description	Source	Description	Destination
order date	Restaurant Database	New order entry	Restaurant Database
order time	Restaurant Database	order items	Restaurant Database
employee id	Restaurant Database	special requests	Restaurant Database
table id	Restaurant Database	order item with special request	
menu item id	Restaurant Database		
quantity	Restaurant Database		
request type	Inventory Database		
request description			

alternative id	Customer Database Customer Database Customer Database		

Major Steps Performed:	**Information for Steps:**
1. INSERT statement to add a new order	Order id (IP)
2. INSERT statement to add order items	Order information (IP)
3. INSERT statement to add special requests	Special request (IP)
4. INSERT statement to link special requests to order items	Order id & special request id (IP)
5. SELECT statement to retrieve order items with special requests	Order information (OP)

Exceptions:

E1: The customer does not order food (occurs at step 1). Therefore, order id will not be created.

Use Case 4: Customer Checkout and Refund Management

Use Case Name: Customer Checkout and Refund Management	**ID Number**: UC-4

Description: This table represents a comprehensive use case that covers all the functionalities related to customer checkout and refund management, including calculating the order total, applying discounts, updating gift card balances, recording tips, and managing refunds if necessary.

Trigger: A customer request

Type: External

Major Inputs:		**Major Outputs:**	
Description	Source	Description	Destination
Order id	Restaurant Database	Order total	Restaurant Database
coupon code	Restaurant Database	discount details	Restaurant Database
coupon id	Restaurant Database	final amount	Customer Database
discount amount	Restaurant Database	updated gift card balance	Employee & Restaurant Database
payer number	Restaurant Database	tip details	
payment method	Customer Database	refund detail	Restaurant Database

single amount card code balance tip amount refund amount refund date reason	Restaurant Database Customer Database Employee & Restaurant Database Customer & Restaurant Database Restaurant Database Restaurant Database		

Major Steps Performed:	**Information for Steps:**
1. Calculate order total 2. Apply discount from coupon (if any) 3. Record payment method and amount 4. Update gift card balance (if used) 5. Record tips for the employee 6. INSERT statement to create a refund record (if necessary) 7. SELECT statements to display final amount, updated gift card balance, tip amount, and refund details	Order information (IP) Coupon information (IP) Payment information (IP) User gift card balance (IP) Tips amount (IP) Refund information (IP) Order information (OP)

Back-end (i.e., reloading inventory, ordering more stocks, signing in and off employees)

Use Case 1: Restaurant Inventory and Supplier Management System

Use Case Name: Restaurant Inventory and Supplier Management System	**ID Number**: UC-1
Description: This table now represents a comprehensive use case that covers all the functionalities of the Restaurant Inventory and Supplier Management System.	
Trigger: A restaurant request supplier replenish inventory **Type**: Temporal & External	

Major Inputs:		**Major Outputs:**	
Description	Source	Description	Destination

Selected Supplier id Selected ingredient id Ingredient id received quantity quantity unit	Supplier Database Restaurant Database Restaurant Database Restaurant Database Restaurant Database Restaurant Database	Supplier name supply type storage condition delivery day contact name phone number email address ingredient name ingredient type ingredient used in cost per unit item name category price replenishment history food inventory quantity	Supplier Database Supplier Database Restaurant Database Supplier Database Supplier Database Supplier Database Restaurant Database Restaurant Database Restaurant Database Restaurant Database Restaurant Database Restaurant Database Restaurant Database Restaurant Database Inventory Database Inventory Database

Major Steps Performed:	**Information for Steps:**
1. SELECT statements to fetch supplier, contact, and ingredient information. 2. INSERT statements to create new replenishment and purchase orders. 3. UPDATE statements to update food inventory with received ingredient quantities.	Suppliers' information (IP) Inventory order updates information (IP) Inventory quantities updates information (IP)

Exceptions:

E1: The ordered products cannot be provided (occurs at step 2). Therefore, inventory will not update.

Use Case 5: Employee Shift Management

Use Case Name: Employee Shift Management	**ID Number**: UC-5
Description: This table represents a comprehensive use case that covers all the functionalities related to employee shift management, including checking employee absence and availability, clocking in and out, calculating hours worked, and managing shift records.	
Trigger: A employee internal activity **Type**: Temporal	

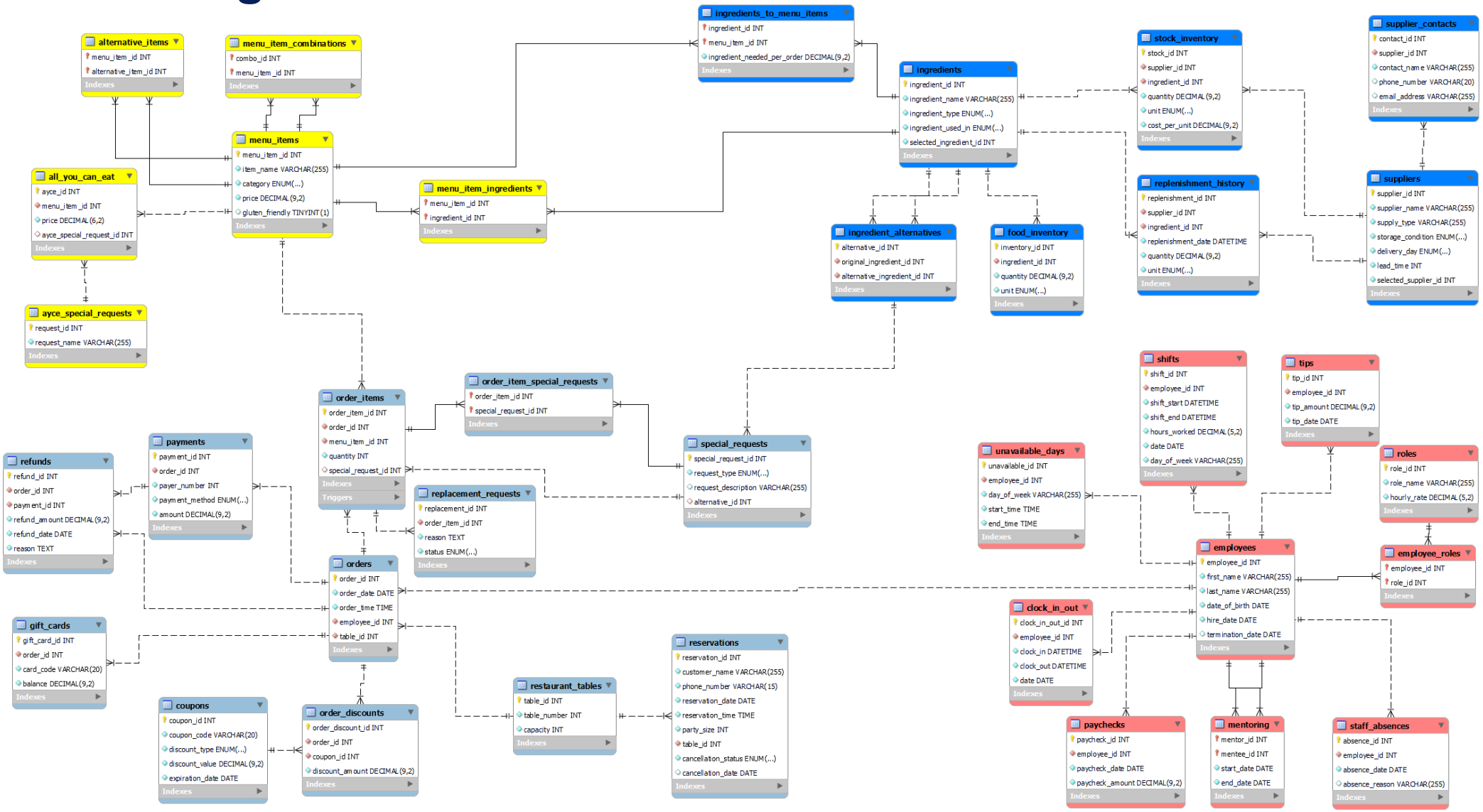
Major Inputs:		**Major Outputs:**	
Description	Source	Description	Destination
Employee id absence date day of week start time end time lock in clock out date	Employee Database Employee Database Employee Database Employee Database Employee Database Employee Database Employee Database Employee Database	Shift details hours worked clock-in and clock-out times	Employee Database Employee Database Employee Database

Major Steps Performed:	**Information for Steps:**
1. Check if an employee is absent on the shift date. 2. Check if an employee has a role assigned. 3. Check if an employee is available for the shift. 4. Clock in at the beginning of a shift 5. Clock out at the end of a shift 6. Calculate hours worked for the shift. 7. Insert a new shift record. 8. Get the shift details before and after the shift	Employee clock in information (OP) Employee role information (OP) Availability (OP) Clock in date (IP) Clock out date (IP) Work hours (IP & OP) Shift record (IP) Employee shift information (OP)

Exceptions:

E1: The employee might not be available for shift day (occurs at step 3). Therefore, the work shift will not be created.

E-R Diagram



Legend

- Menu Section
- Supply Section
- Order Section
- Employee Section

Data Insertion & Assumptions

When inserting data, we performed online and physical research to include real public data about the following in our database:

- Menu items
- Coupon discounts
- Employee roles
- Employee Shifts
- All you can eat and special requests
- Gift cards starting amounts
- Typical special requests (allergies, etc.)
- Restaurant tables
- Stock inventory (Only items)

However, we also made **assumptions** when adding data where data was not publicly available, including:

- Orders: We created a small random set of order dates and times within C-Lover's hours, and randomly assigned the order status and customers to it.
- Order items: We randomly selected menu items for each order, and added special requests based on what a typical request would be.
- Payments: For each payment, we generated a random payment method and assumed that people were reasonable when ordering food (usually under $200). We also created random values for current gift card balances and assumed they paid within normal hours.
- Employees: Created a random set of employee's names, their role, and their duration with C-Lovers as well as other employee information.
- Employee Work Schedules: Randomly assigned employee schedules during operations reasonably.
- Paychecks and Tips: Randomly assigned reasonable amounts to pay employee and customers to tip them.

- Supplier info: We created fake contacts and supplier information, but the items they were delivering are really physically used in the restaurant.
- Unavailable days: Assumed reasonable unavailable days for employees.
- Replenishment History: Randomly assigned reasonable times and information for fulfilling the replenishment history of C-Lovers
- Replacement Requests: Randomly assigned unique special requests that a restaurant may see.
- Reservations: Assumed random names and customer information to create reasonable reservations and C-Lovers.
- Coupons: Created coupons in case C-Lovers create percentage based or fixed amount coupons and generated random reasonable values.
- Staff Absences: Created typical reasons randomly of why an employee missed work.

See “C-Lovers SQL Database” Doc or Appendix 5 to see the SQL queries.

SQL Queries for Use Cases

We performed SQL queries for all use cases we envisioned in our ERD and adjusted if the code was not working. To see the SQL Queries in further depth, open the “C-Lovers SQL database” and “SQL Queries and 10 Report Questions” Doc on Appendix 5 and 6.

Reporting Queries (10 Questions)

We created 10 possible questions that a manager may want to see and included it in our report, including:

1. What were the total sales across all orders for today (orders, payments, giftcards, discounts, all_you_can_eat)?
2. At what times during the day are the most orders being placed and least orders being placed (Maybe 1 hour intervals through the day)?
3. Which employee made the most tips in this month (April, 2023)?
4. What is the total amount paid for orders made on a specific date? What about any day?
5. How many employees have the "Manager" role?
6. What is the total number of hours worked by all employees on a specific day?
7. Which menu items have ingredients that are stored in the "frozen" condition?
8. What is the total number of hours worked by each employee?
9. Which employees mentored another employee for the longest period of time?
10. List all orders with the customer name, order date, menu item name, and price for each menu item.

See “C-Lovers SQL Database” and “SQL Queries and 10 Report Questions” Doc or Appendix 5 and 6 to see the SQL queries.

Reflection

To create a legitimate MySQL database modeling a real-world restaurant, we first set criteria on which restaurant to choose: 1. A franchise so it's easily visitable, 2. Low food customization, 3. BC-operated and easily contactable. Thus, we settled on C-Lovers Fish & Chips (mostly their Coquitlam location), using their website and observing their operations to construct our database, aided by AI-assisted programming.

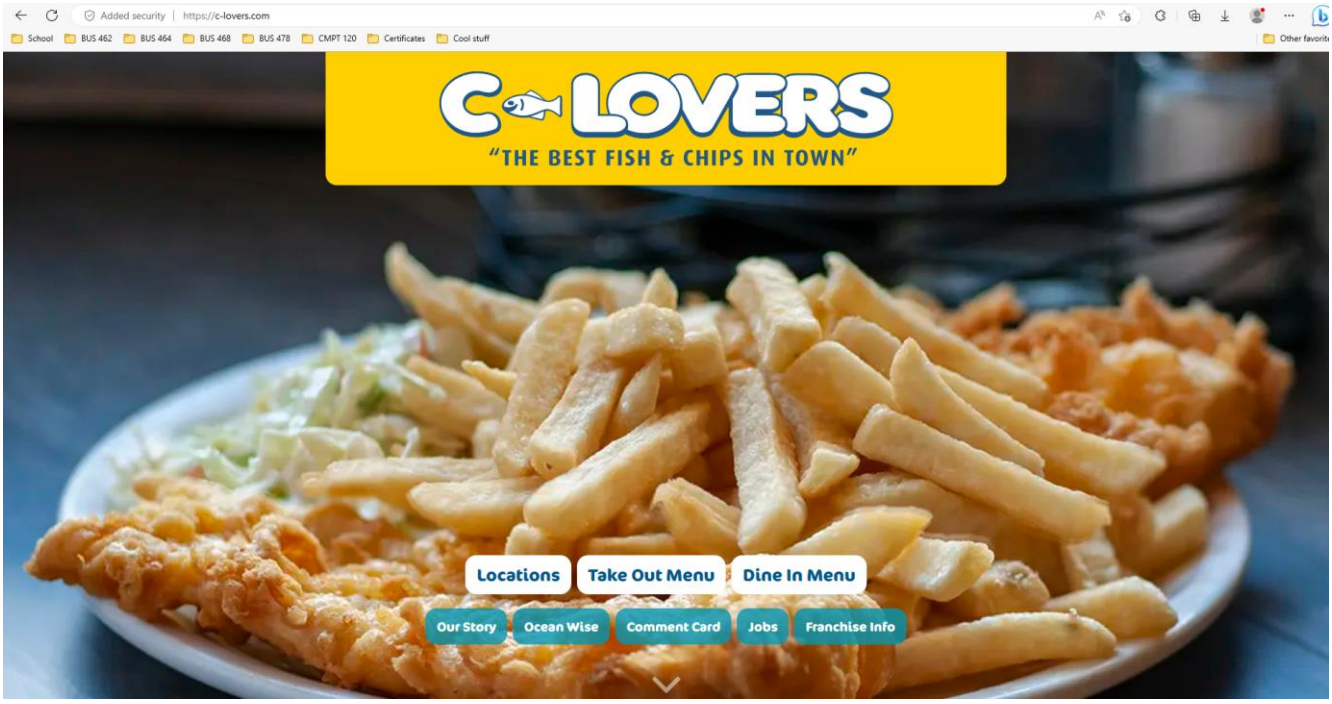
Initially, we split the work by Tianci researching the website while Jaiden studied the restaurant processes and asked the manager questions. Then, combining this information, we created SQL tables representing C-Lover's business processes, developed use cases to connect the tables, and constructed an ERD diagram. GitHub Copilot was critical in defining table attributes, writing foreign and primary keys, and suggesting relevant tables through SQL coding for our use cases. We also used ChatGPT to verify code correctness and factuality with C-Lovers, ensuring a comprehensive solution.

After adding sample and real data to the schema, we executed SQL query for the use cases, then formulated 10 report questions that managers would ask and executed those queries. When facing issues during this, we refined the code and ERD as needed to address and execute the queries, such as missing or incorrect table relationships.

In conclusion, this assignment requires strong critical thinking and analysis to investigate a restaurant business and create its SQL representation, even with the assistance of AI. Throughout everything, we faced difficulties like translating our information to SQL, code not working properly, or complicated structures like recursives. Luckily, we used example restaurant ERDs, online support, and AI assistance to help shape what our project looked like. Though this was a strenuous process, it was fun collaborating with AI on this field research project, and we hope that future projects can incorporate AI alongside human problem-solving in the workplace.

Appendix

Appendix 1: C-Lovers Website



Appendix 2: C-Lovers Coquitlam Outside View (Google Maps)

Appendix 3: C-Lovers Menu Front

Appendix 4: C-Lovers Menu Back

Appendix 5: C-Lovers Database Link

https://drive.google.com/file/d/1SC52pWlSAzbHDmtf3_cq5odfJ3PKSlc3/view?usp=share_link

Appendix 6: C-Lovers SQL Queries and 10 Report Questions Link

https://drive.google.com/file/d/1PkUYxRfvOSWUoJdslF-jAB-1pAYwlU4J/view?usp=share_link

Appendix 7: C-Lovers Slides Presentation

https://drive.google.com/file/d/1GfqbLXjNSlDR7gfgyfJVrj3wuJm8dgGT/view?usp=share_link