Project 3:

Logical Database Design

Written By

Mr.   Suppakorn Pojsomphong   6588077

Miss Nuttida   Wiphaalongkot 6588144
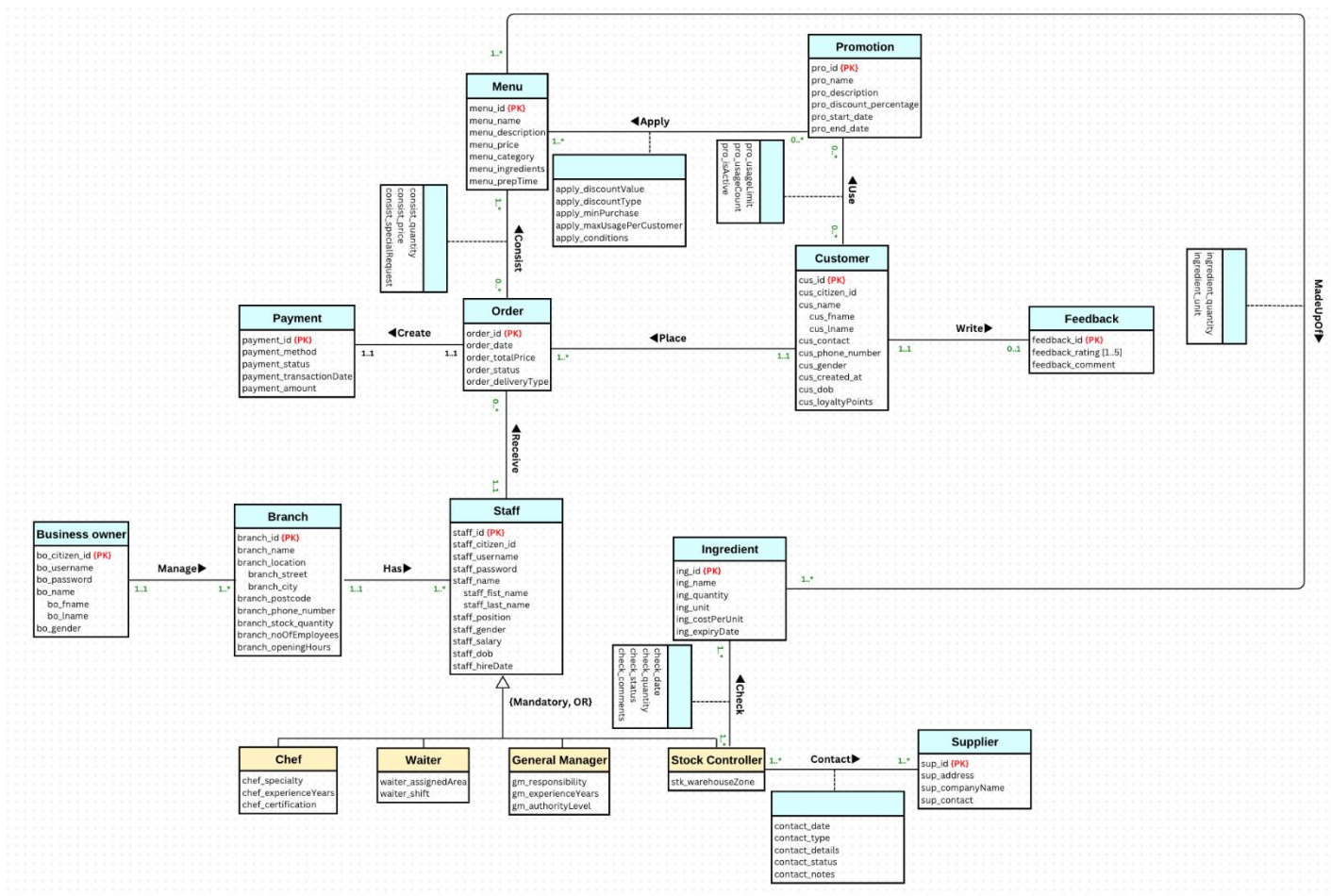
Miss Ananya   Sangtunchai   6588191

A Report Submitted in Partial Fulfillment of

the Requirements for

ITCS413 Database Design

Faculty of Information and Communication Technology

Semester 2/2024

Table of Contents

LOGICAL DATA MODEL

# DATA DICTIONARY

Entity	Attributes
Business Owner	`bo_citizen_id {PK}` – Unique identifier for business owner `bo_username` – Username of business owner `bo_password` – Password of business owner `bo_name` `bo_fname` – First name of business owner `bo_lname` – Last name of business owner `bo_gender` – Male/Female of business owner
Branch	`branch_id {PK}` – Unique identifier for each branch `branch_name` – Each branch must have a unique name `branch_street` – street of the branch (e.g. Park Avenue, Oak Street) `branch_city` – city of the branch (e.g. New York City, Las Vegas) `branch_postcode` – postcode of the branch (e.g. 10001, 30301, 48201) `branch_phone` – Contact number of the branch `branch_stockQuantity` – Total amount of stock available at the branch `branch_noOfEmployees` – Number of employees working at the branch. `branch_openingHours` – Operating hours of the branch. `bo_citizen_id {FK}` – Foreign key linking specific branch to a specific business owner.
Staff	`staff_id {PK}` – Unique identifier for each staff `staff_citizenID` – National identification number `staff_username` – Username of staff `staff_password` – Password of staff `staff_name` `staff_fname` – First name of staff `staff_lname` – Last name of staff `staff_email` – Email of staff `staff_gender` – Male/Female of staff `staff_salary` – Monthly salary of staff `staff_dob` – Useful for internal records and HR purposes. `staff_hireDate` – Timestamp when the staff was added `branch_id {FK}` – Foreign key linking staff to their assigned branch.
Chef *(Subclass of Staff)*	`staff_id {PK, FK}` – Unique identifier for each staff `chef_specialty` – Area of expertise (e.g. Pastry, Grilled Dishes) `chef_experienceYears` – Number of years of experience (e.g., 5 years in Pastry). `chef_certification` – If certifications like "Certified Pastry Chef" matter.
Waiter *(Subclass of Staff)*	`staff_id {PK, FK}` – Unique identifier for each staff `waiter_assignedArea` – Assigned area in the restaurant (e.g. Indoor, Outdoor) `waiter_shift` – Morning, Afternoon, Night (use ENUM).

General Manager *(Subclass of Staff)*	**staff_id {PK, FK}** – Unique identifier for each staff **gm_responsibility** – Department managed (e.g. Kitchen, Customer Service) **gm_experienceYears** – Years of experience in management. **gm_authorityLevel** – Permissions or decision-making powers (use ENUM, such as Full, Partial).
Stock Controller *(Subclass of Staff)*	**staff_id {PK, FK}** – Unique identifier for each staff **sc_warehouseZone** – The stock area managed
Menu	**menu_id {PK}** – Unique identifier for each menu item **menu_name** – Name of the dish **menu_description** – Brief description **menu_price** – Selling price **menu_category** – Type of food (e.g. Appetizer, Main Course, Dessert) **menu_prepTime** – Estimated preparation time.
Customer	**cus_id {PK}** – Unique identifier for each customer **cus_citizenID** – National identification number **cus_fname** – First name **cus_lname** – Last name **cus_email** – Email or alternative contact **cus_phone** – Phone number **cus_gender** – Male/Female/Other **cus_created_at** – Timestamp when the customer was added **cus_dob** – Date of birth (for customer segmentation or special promotions).
Order	**order_id {PK}** – Unique identifier for each order **order_date** – Date and time of the order **order_totalPrice** – Final total cost of an entire order. **order_status** – Order status (Pending, Completed, Canceled) **order_deliveryType** – Type of order (e.g. Dine-in, Takeout, Delivery)
Consist of	**order_id (FK, PK)** – Foreign key linking to the order **menu_id (FK, PK)** – Foreign key linking to the menu item **consist_quantity** – Number of times this menu item appears in the order (e.g., 2 pizzas) **consist_price** – total price for each menu item in the order. **consist_specialRequest** – Any custom request by the customer (e.g., "No onions")
Payment	**payment_id {PK}** – Unique identifier for each payment **payment_method** – Method of payment (e.g. Credit Card, Cash, Mobile Banking) **payment_status** – Status of payment transaction (e.g. Pending, Completed, Failed) **payment_transactionDate** – Timestamp **payment_amount** – To store the actual payment amount. **order_id** – Foreign key linking payment to a specific order.
Feedback	**feedback_id {PK}** – Unique identifier for each feedback **feedback_rating** – 1 to 5 stars

	• **feedback_comment** – Customer's comment
Promotion	• **pro_id {PK}** – Unique identifier for each promotion • **pro_name** – Promotion title • **pro_description** – Details of the promotion • **pro_discounPercentage** – Discount applied • **pro_startDate** – Promotion start date • **pro_endDate** – Promotion end date • **pro_usageLimit** – The maximum number of times a promotion can be used. • **pro_usageCount** – Tracks how many times the promotion has been used. • **pro_isActive** – Helps indicate whether a promotion is still available (Boolean: TRUE or FALSE).
Apply	• **menu_id (PK, FK)** – Foreign key referencing the Menu entity (the menu item that the promotion applies to). • **pro_id (PK, FK)** – Foreign key referencing the Promotion entity (promotion applied to the menu item). • **apply_discountValue** – The actual discount amount applied to this menu item. • **apply_discountType** – Type of discount: "Percentage", "Fixed Amount", "Buy X Get Y Free". • **apply_minPurchase** – Minimum purchase amount required to activate the promotion. • **apply_maxUsagePerCustomer** – Maximum times an individual customer can use this promotion. • **apply_conditions** – Special conditions (e.g., "Valid only on weekends").
Use	• **pro_id {PK, FK}** – Foreign key from Promotion. • **cus_id {PK, FK}** – Foreign key from Customer. • **use_usageCount** – Tracks how many times this customer has used the promotion. • **use_lastUsed** – Timestamp of the last time this customer used the promotion. • **use_isEligible** – Boolean flag (TRUE = eligible, FALSE = not eligible).
Supplier	• **sup_id (PK)** – system-generated ID • **sup_address** – Address of supplier • **sup_companyName** – Supplier company name • **sup_contact** – Contact of supplier
Contact	• **staff_id (PK,FK)** – Represents the Stock Controller involved in the communication. • **sup_citizenID (PK, FK)** – Represents the Supplier involved in the communication. • **contact_date** – When the communication took place. • **contact_type** – Type of interaction (Email, Phone, Meeting, etc.). • **contact_details** – Key information about the contact. • **contact_status** – Helps track the progress/status of the contact. • **contact_notes** – Additional helpful information, like follow-up actions, agreements, or specific supplier needs.
Ingredient	• **ing_id {PK}** – Unique identifier for each ingredient • **ing_name** – Name of the ingredient • **ing_quantity** – Stock quantity available

	• **ing_unit** – Measurement unit (e.g. kg, liters, pieces, etc.) • **ing_costPerUnit** – Cost per unit • **ing_expiryDate** – To track freshness and prevent waste. • **ing_supplierID** – Foreign key linking to Supplier.
Check	• **staff_id {PK, FK}** – Foreign key linking to the Stock Controller who is responsible for checking the ingredient. • **ing_id {PK, FK}** – Foreign key linking to the ingredient being checked by the stock controller. • **check_date** – The date and time when the ingredient was checked. • **check_quantity** – The quantity of the ingredient checked. • **check_status** – Status of the check (e.g., Completed, Pending, Discrepancy). • **check_comments** – Additional notes or comments related to the check (e.g., any discrepancies found, special observations, etc.).
MadeUpOf	• **menu_id {PK, FK}** – Foreign key referring to the Menu entity (the dish being made). • **ing_id {PK, FK}** – Foreign key referring to the ingredient entity (the ingredient used in the dish). • **ingredient_quantity** – Quantity of the ingredient used in the dish (e.g., 100g of tomatoes, 2 slices of bread). • **ingredient_unit** – Measurement unit for the ingredient quantity (e.g., grams, pieces, cups).

RELATIONAL DATABASE SCHEMA
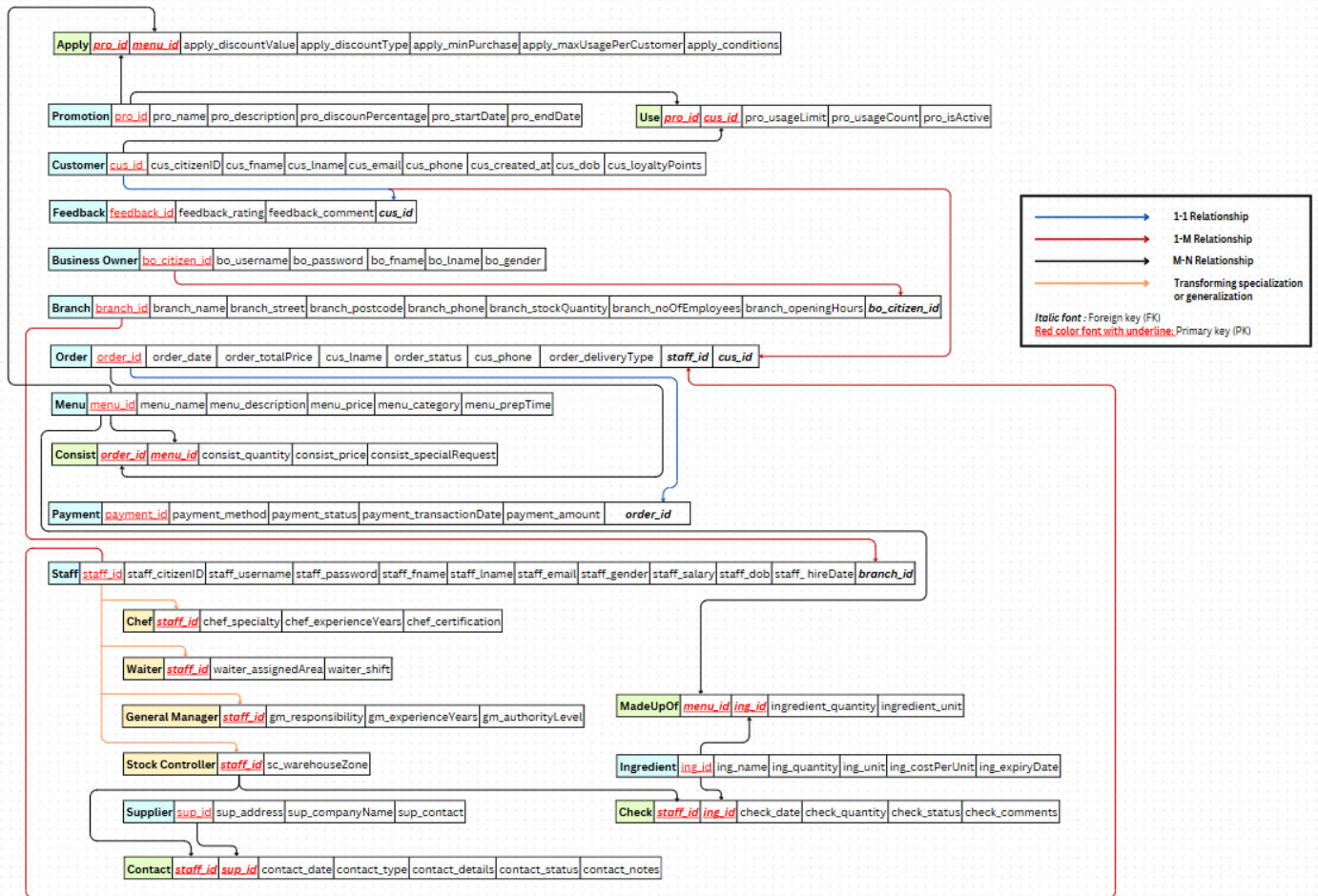
https://www.canva.com/design/DAGhHnQ5ZTo/PqykRRQULkOmhp4q7SFDwA/edit?utm_content=DAGhH
nQ5ZTo&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton

ENTITY RELATIONSHIPS AND CARDINALITIES

Entities Involved	Relationship	Cardinality	Constraint	Description
Business Owner - Branch	Manage [1:M]	[1:M]	(1,M)	Each business owner can manage at least one branch.
		[1:1]	(1,1)	Each branch can be managed by at most one business owner.
Branch - Staff	Has [1:M]	[1:M]	(1,M)	Each branch can have at least one staff member.
		[1:1]	(1,1)	Each staff member can belong to at most one branch.
Staff - Order	Receive [1:M]	[1:M]	(0,M)	Each staff member may or may not receive orders.
		[1:1]	(1,1)	Each order can be received by exactly one staff member.
Order - Menu	Consist [M:N]	[1:M]	(1,M)	Each order can consist of at least one menu item.
		[1:M]	(0,M)	Each menu item may or may not belong to orders.
Order - Payment	Create [1:1]	[1:1]	(1,1)	Each order can have at least one payment.
		[1:1]	(1,1)	Each payment can belong to exactly one payment.
Customer - Order	Place [1:M]	[1:M]	(1,M)	Each customer can place at least one order.
		[1:1]	(1,1)	Each order can be placed by exactly one customer.
Customer - Feedback	Write	[1:M]	(0,M)	Each customer may or may not submit feedback.

		[1:M]	[1:1]	(1,1)	Each feedback can be submitted by exactly one customer.
Customer - Promotion	Use [M:N]	[1:M]	(0,M)	Each customer may or may not use multiple promotions.
		[1:M]	(0,M)	Each promotion may or may not be used by multiple customers.
Stock Controller - Ingredient	Check [M:N]	[1:M]	(0,M)	Each stock controller may or may not check ingredients.
		[1:N]	(0,N)	Each ingredient may or may not be checked by stock controllers.
Stock Controller - Supplier	Contact	[1:M]	(1,M)	Each stock controller can contact at least one supplier.
		[1:N]	(1,N)	Each supplier
Menu - Ingredient	Made up of	[1:M]	(1,M)	Each many can be made up of 1 or many ingredients.
		[1:N]	(1,N)	Each ingredient can be in 1 or many menus.
Menu - Promotion	Apply	[1:M]	(0,M)	Each menu can be applied by many promotions or none at all.
		[1:N]	(1,N)	Each promotion can apply in 1 or many menus.

EDITING CHECKLISTS

1. Additions and Modifications:

1. Branch	**Add:** • `branch_openingHours` – Operating hours of the branch.
2. Customer	**Modify:** • `cus_contact → cus_email` – To clearly indicate that it stores email. **Add:** • `cus_dob` – Date of birth (for customer segmentation or special promotions). • `cus_loyaltyPoints` – If implementing a loyalty program.
3. Staff (Superclass)	**Modify:** • `staff_created_at → staff_hireDate` – More meaningful in tracking employment history. **Add:** • `staff_salary` – Monthly salary. • `staff_branchID` – Foreign key linking staff to their assigned branch. • `staff_dob` – Useful for internal records and HR purposes.
4. Business Owner	**Modify:** • `bo_citizen_id (PK) → bo_id (PK)` – Instead of using the national ID, use an auto-generated identifier. • `bo_username → bo_email` – Typically, usernames are replaced by emails.
5. Ingredient	**Modify:** • `ing_price → ing_costPerUnit` – To avoid confusion with the selling price of the ingredient. **Add:** • `ing_expiryDate` – To track freshness and prevent waste. • `ing_supplierID` – Foreign key linking to Supplier.
6. Menu	**Add:**

	• `menu_ingredients` – List of ingredients required for the dish. • `menu_prepTime` – Estimated preparation time.
7. Order	**Modify:** • `order_deliveryType` → `order_type` – Covers dine-in, takeout, and delivery. **Add:** • `order_staffID` – Foreign key linking to the waiter/chef handling the order. • `order_branchID` – Foreign key linking to the branch where the order was placed.
8. Payment	**Add:** • `order_id (FK)` – Foreign key linking payment to a specific order. • `payment_amount` – To store the actual payment amount.
9. Feedback	**Add:** • `cus_id (FK)` – Links feedback to a specific customer. • `menu_id (FK)` – Links feedback to a specific menu item.
10. Promotion	**Add:** • `menu_id (FK)` – Links promotion to specific menu items.
11. Supplier	**Modify:** • `sup_citizenID (PK)` → `sup_id (PK)` – Use a system-generated ID instead of a national ID. **Add:** • `sup_companyName` – If the supplier is a company instead of an individual.
12. Chef (Subclass)	**Add:** • `chef_experienceYears` – Number of years of experience (e.g., 5 years in Pastry). • `chef_certification` – If certifications like "Certified Pastry Chef" matter.
13. Waiter (Subclass)	**Modify:** • `section` → `waiter_assignedArea`.

	Add: • `waiter_shift` – Morning, Afternoon, Night (use ENUM).
14. General Manager (Subclass)	**Add:** • `gm_experienceYears` – Years of experience in management. • `gm_authorityLevel` – Permissions or decision-making powers (e.g., Full, Partial).
15. Stock Controller (Subclass)	**Modify:** • Change entity name to **Stock Controller** • `warehouse_section` → `inv_manager_warehouseZone` (More descriptive). **Add:** • `inv_manager_inventorySystem` – System used (e.g., Manual, Digital).

2. Create more Many-to-Many relationship between Promotion and Customer called "Use"

Relationship:

- Each customer may or may not use multiple promotions.
- Each promotion may or may not be used by multiple customers.

Attributes:

- **`pro_id {PK, FK}`**– Foreign key from Promotion.
- **`cus_id {PK, FK}`** **–** Foreign key from Customer.
- **`use_usageCount`** **–** Tracks how many times this customer has used the promotion.
- **`use_lastUsed`** **–** Timestamp of the last time this customer used the promotion.
- **`use_isEligible`** **–** Boolean flag (TRUE = eligible, FALSE = not eligible).

Clarification of `use_isEligible` in the Use Table

- **Definition:** `use_isEligible` is a Boolean attribute (TRUE/FALSE) that determines whether a customer is eligible to use a specific promotion.
- **Why is `use_isEligible` Useful?**
  1. Prevent invalid promotions from being applied automatically.
  2. Speed up the checkout process by checking eligibility in advance
  3. Ensure customers follow the promotional rules

- **Conditions for `use_isEligible` = TRUE:**

A customer is eligible to use a promotion if all of the following conditions are met:

1. **Promotion is active (`pro_isActive` = TRUE)**
   o The promotion is still available and has not expired (`pro_endDate` has not passed).
2. **Promotion has not exceeded its total usage limit**
   o `pro_usageCount < pro_usageLimit`
3. **Customer has not exceeded their individual usage limit**
   o `use_usageCount < apply_maxUsagePerCustomer` (if applicable)
4. **Order meets the promotion's minimum purchase requirement**
   o The total price of the order is ≥ `apply_minPurchase` (if required)
5. **The promotion applies to at least one menu item in the order**
   o There is a matching record in the Apply Table between the selected promotion and the menu items in the order
6. **Customer meets any special conditions**
   o Example: "New customers only" (if `apply_conditions` specify it)

Example Scenarios

Eligible (`use_isEligible` = TRUE)

Customer	Promotion	Usage Count	Usage Limit	Max Usage per Customer	Order Total	Min Purchase	Active?	Result
C001	P001 (10% off on burgers)	3	100	5	$20	$15	TRUE	Eligible
C002	P002 (Buy 1 Get 1 Free Fries)	0	50	1	$10	$10	TRUE	Eligible

Not Eligible (`use_isEligible` = FALSE)

Customer	Promotion	Usage Count	Usage Limit	Max Usage per Customer	Order Total	Min Purchase	Active?	Reason	Result
C003	P001 (10% off on burgers)	100	100	5	$20	$15	TRUE	**Usage Limit Reached**	Not Eligible
C004	P002 (Buy 1 Get 1 Free Fries)	1	50	1	$10	$10	TRUE	**Max Usage Per Customer Reached**	Not Eligible
C005	P003 ($5 off orders above $30)	2	200	3	$25	$30	TRUE	**Order Total < Min Purchase**	Not Eligible
C006	P004 (Special for New Customers)	0	50	1	$20	$15	TRUE	**Not a New Customer**	Not Eligible

3. Create more Many-to-Many relationship between Menu and Ingredient called "MadeUpOf"

Relationship:

- Each customer may or may not use multiple promotions.
- Each promotion may or may not be used by multiple customers.

Attributes:

- **menu_id (PK, FK)** – Foreign key referring to the Menu entity (the dish being made).
- **ing_id (PK, FK)** – Foreign key referring to the ingredient entity (the ingredient used in the dish).
- **ingredient_quantity** – Quantity of the ingredient used in the dish (e.g., 100g of tomatoes, 2 slices of bread).
- **ingredient_unit** – Measurement unit for the ingredient quantity (e.g., grams, pieces, cups).
- **ingredient_cost** – The cost of the ingredient in the specific quantity for the menu item. This can be useful to calculate the total cost of the menu item.

Example of "MadeUpOf" Entity:

menu_id	ing_id	ingredient_quantity	ingredient_unit	ingredient_cost
1	101	2	slices	0.5
1	102	100	grams	0.3
2	103	1	piece	0.7

- After that, remove menu_ingredients out of Menu entity

4. Create more Many-to-Many relationship between Menu and Promotion called "Apply"

Relationship:

- Each menu can be applied by many promotions or none at all.
- Each promotion can apply in 1 or many menus.

Attributes:

- **menu_id (PK, FK)** – Foreign key referencing the Menu entity (the menu item that the promotion applies to).
- **pro_id (PK, FK)** – Foreign key referencing the Promotion entity (promotion applied to the menu item).
- **apply_discountValue** – The actual discount amount applied to this menu item.
- **apply_discountType** – Type of discount: "Percentage", "Fixed Amount", "Buy X Get Y Free".
- **apply_minPurchase** – Minimum purchase amount required to activate the promotion.
- **apply_maxUsagePerCustomer** – Maximum times an individual customer can use this promotion.
- **apply_conditions** – Special conditions (e.g., "Valid only on weekends").

Promotion	Menu	Discount Type	Discount Value	Min Purchase	Max Usage Per Customer	Conditions
P001 (Summer Deal)	Burger	Percentage	10%	None	2	Only for classic burgers
P001 (Summer Deal)	Drink	Percentage	5%	None	2	Soft drinks only
P002 (Buy 1 Get 1 Free)	Large Fries	Buy X Get Y Free	100%	Buy 1	1	Only valid for large size

If we put these attributes in the Promotion table, we would not be able to apply different rules for different menus.

REFERENCES

Panmore Institute. (n.d.). *Burger King's organizational structure (An analysis)*. Panmore. Retrieved January 24, 2025, from https://panmore.com/burger-king-organizational-structure-analysis

Burger King. (n.d.). *Burger King application*. Retrieved February 1, 2025, from https://apps.apple.com/th/app/burger-king-thailand/id1440299482?l=th

Burger King. (n.d.). *Burger King website*. Retrieved February 3, 2025, from https://www.burgerking.com