LION BITES

Database Design Document

Version 1.0

Prepared by:

Leon Marco A. Devela

Table of Contents

I. Introduction

This document is the Database Design Document (DDD) for Lion Bites, a food ordering application inspired by the GrabFood app. Developed as part of an Information Management course project for the Bachelor of Science in Computer Science with specialization in Machine Learning program, this document outlines the overall database architecture and design specifications for Lion Bites.

I.I Document Objectives

The objectives of this Database Design Document (DDD) for Lion Bites are to:

- Detail the MySQL Database Design: Outline the structure and organization of the MySQL database, explaining how data is stored in tables and accessed by users or programs via a database management system (DBMS).
- Provide Comprehensive Documentation: Include an Entity Relationship Diagram (ERD) that visually represents the database structure, a Data Dictionary detailing data elements, clear descriptions of the purpose of each table, and an explanation of the relationships among the tables.
- Guide Implementation and Maintenance: Serve as a blueprint for the database implementation, offering clear insights into the design that will aid developers, testers, and support teams in ensuring that the system meets its requirements and can be effectively maintained and enhanced over time.

I.II Intended Audience

This document is intended for:

- Technical Reviewers and Academic Supervisors: Including course supervisors and university faculty responsible for evaluating the quality and rigor of this project documentation.

- Lion Bites Development Team:
    - Architects: Tasked with ensuring the overall system architecture meets the project's requirements.
    - Designers: Responsible for crafting system designs that align with the specifications outlined in this document.
    - Programmers: Charged with implementing the database and integrating it with the overall application as detailed in this DDD.
    - Testers: Who will validate that the implemented system meets all specified requirements through thorough testing.

I.III References

This document refers to the following references:

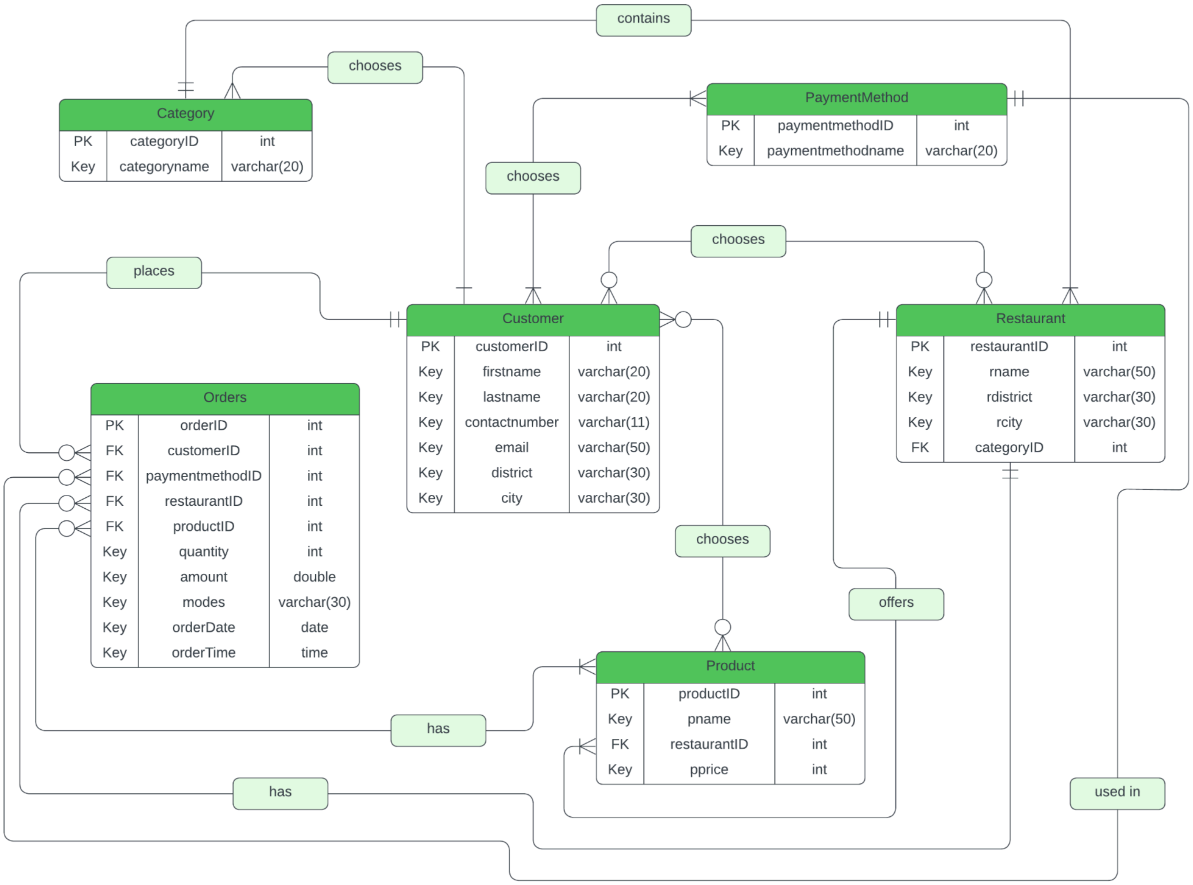
- CTINFMGL Project Specifications: https://believed-bongo-319.notion.site/CTINFMGL-Project-Specifications-19296450aad180aea9ebf27987415f4c

II. Detailed Database Design

This section outlines the design of the MySQL database for Lion Bites and includes the following key components:

1. **Entity Relationship Diagram (ERD)**: A visual representation of the database schema that illustrates the tables and their interrelationships.
2. **Data Dictionary**: A comprehensive listing of all data elements—including definitions, data types, and constraints—that serves as a reference for the database structure.
3. **Purpose of Each Table**: A detailed explanation of the role and function of each table within the database, clarifying how each contributes to the overall functionality of Lion Bites.
4. **Relations**: An outline of the relationships between tables, including primary and foreign key associations, which ensures data integrity and supports efficient data retrieval.

II.I Entity Relationship Diagram



Category		
PK	categoryID	int
Key	categoryname	varchar(20)

PaymentMethod		
PK	paymentmethodID	int
Key	paymentmethodname	varchar(20)

Customer		
PK	customerID	int
Key	firstname	varchar(20)
Key	lastname	varchar(20)
Key	contactnumber	varchar(11)
Key	email	varchar(50)
Key	district	varchar(30)
Key	city	varchar(30)

Restaurant		
PK	restaurantID	int
Key	rname	varchar(50)
Key	rdistrict	varchar(30)
Key	rcity	varchar(30)
FK	categoryID	int

Orders		
PK	orderID	int
FK	customerID	int
FK	paymentmethodID	int
FK	restaurantID	int
FK	productID	int
Key	quantity	int
Key	amount	double
Key	modes	varchar(30)
Key	orderDate	date
Key	orderTime	time

Product		
PK	productID	int
Key	pname	varchar(50)
FK	restaurantID	int
Key	pprice	int

Figure 1: Entity Relationship Diagram

"Lion Bites" is designed to facilitate customers placing orders for products offered by various restaurants. The system organizes restaurants by categories, allows customers to select payment methods, and tracks detailed order information, including products, quantities, and delivery or pickup preferences. It connects customers, restaurants, products, and payment methods through a series of well-defined relationships.

II.II Data Dictionary

A data dictionary provides a detailed description of each attribute in a database, including its name, data type, constraints, and purpose. Below is the data dictionary for the "Lion Bites" system, organized by table, based on the provided attribute information.

II.II.I Data Dictionary for entity: Category

Attribute Name	Data Type	Constraint	Description
categoryID	int	AUTO_INCREMENT, PRIMARY KEY	A unique identifier for each category, used as the primary key in the Category table.
categoryname	varchar(20)	NOT NULL, UNIQUE	The name of the category, limited to 20 characters, describing the type of product or service.

II.II.II Data Dictionary for entity: PaymentMethod

Attribute Name	Data Type	Constraint	Description
paymentmethodID	int	AUTO_INCREMENT, PRIMARY KEY	A unique identifier for each payment method, used as the primary key in the PaymentMethod table.
paymentmethodname	varchar(20)	NOT NULL, UNIQUE	The name of the payment method, limited to 20 characters, indicating the type of payment used.

II.II.III Data Dictionary for entity: Restaurant

Attribute Name	Data Type	Constraint	Description
restaurantID	int	AUTO_INCREMENT, PRIMARY KEY	A unique identifier for each restaurant, used as the primary key in the Restaurant table.
rname	varchar(50)	NOT NULL, UNIQUE	The name of the restaurant, limited to 50 characters.
rdistrict	varchar(30)	NOT NULL, UNIQUE	The district where the restaurant is located, limited to 30 characters.
rcity	varchar(30)	NOT NULL, UNIQUE	The city where the restaurant is located, limited to 30 characters.
categoryID	int	NOT NULL	Foreign key referencing the categoryID in the Category table, linking a restaurant to a category..

II.II.IV Data Dictionary for entity: Customer

Attribute Name	Data Type	Constraint	Description
customerID	int	AUTO_INCREMENT, PRIMARY KEY	A unique identifier for each customer, used as the primary key in the Customer table.
firstname	varchar(20)	NOT NULL	The first name of the

			customer, limited to 20 characters.
lastname	varchar(20)	NOT NULL	The last name of the customer, limited to 20 characters.
contactnumber	varchar(11)	NOT NULL, UNIQUE, REGEXP '^[0-9]{1,11}$'	The contact number of the customer, limited to 11 characters, typically a phone number.
email	varchar(50)	NOT NULL, UNIQUE	The email address of the customer, limited to 50 characters, used for communication.
district	varchar(30)	NOT NULL	The district where the customer resides, limited to 30 characters.
city	varchar(30)	NOT NULL	The city where the customer resides, limited to 30 characters.

II.II.V Data Dictionary for entity: Orders

Attribute Name	Data Type	Constraint	Description
orderID	int	AUTO_INCREMENT, PRIMARY KEY	A unique identifier for each order, used as the primary key in the Orders table.
customerID	int	NOT NULL	Foreign key referencing the customerID in the Customer table,

			linking an order to a customer.
paymentmethodID	int	NOT NULL	Foreign key referencing the paymentmethodID in the PaymentMethod table, linking an order to a payment method.
restaurantID	int	NOT NULL	Foreign key referencing the restaurantID in the Restaurant table, linking an order to a restaurant.
productID	int	NOT NULL	Foreign key referencing the productID in the Product table, linking an order to a product.
quantity	int	NOT NULL	The number of items ordered, an integer representing the quantity of a product in an order.
amount	double	NOT NULL	The total amount for the order, a decimal number representing the cost.
mode	varchar(30)	NOT NULL	The mode of the order (e.g., delivery, pickup), limited to 30 characters.
orderDate	date	NOT NULL	The date when the order was placed, stored in date

			format.
orderTime	time	NOT NULL	The time when the order was placed, stored in time format.

II.II.VI Data Dictionary for entity: Product

Attribute Name	Data Type	Constraint	Description
productID	int	AUTO_INCREMENT, PRIMARY KEY	A unique identifier for each product, used as the primary key in the Product table.
pname	varchar(50)	NOT NULL	The name of the product, limited to 50 characters, describing the item being sold.
restaurantID	int	NOT NULL	Foreign key referencing the restaurantID in the Restaurant table, linking a product to a restaurant.
price	int	NOT NULL	The price of the product, an integer representing the cost in currency units.

II.III Purpose of each tables

The "Purpose of Each Table" section offers a comprehensive overview of the role and significance of every table within the "Lion Bites" restaurant order system. Each table is meticulously designed to store specific types of data critical for managing restaurant operations, including customer details, order information, product listings, payment methods, and restaurant categories. Below is the explanation of the purpose of each table in the "Lion Bites" system, organized by table, based on the provided schema information, to illustrate how these tables collectively support the efficient management of restaurant-related data and ensure seamless functionality of the platform.

II.III.I Purpose of Category Table

This table stores information about the different product categories available in the "Lion Bites" system. The categoryID is a unique identifier for each category, and the categoryname (varchar(20)) contains the name of the category, such as "Fast food," "Chinese," or "Cafe." The purpose of this table is to organize products into specific groups, making it easier to manage and retrieve them. It connects to the Restaurant table through the categoryID foreign key, ensuring that products offered by restaurants are linked to appropriate categories. This enables efficient categorization and filtering of products based on restaurant offerings.

II.III.II Purpose of PaymentMethod Table

This table stores details about the payment methods available to customers in the "Lion Bites" system. The paymentmethodID uniquely identifies each payment method, and the paymentmethodname (varchar(20)) holds the name, such as "Credit Card," "Cash," or "Online Payment." The purpose of this table is to manage and track the payment options customers can select. It links to the Orders table through the paymentmethodID foreign key, ensuring each order is associated with the chosen payment method.

II.III.III Purpose of Restaurant Table

This table stores information about the restaurants participating in the "Lion Bites" system. The restaurantID uniquely identifies each restaurant, and fields like name (varchar(50)), district (varchar(30)), city (varchar(30)), and categoryID (foreign key) include the restaurant’s name, location, and category. Its purpose is to maintain a directory of restaurants, their locations, and associated categories, connecting to the Category table via the categoryID foreign key, the Orders table via the restaurantID foreign key, and the Product table to tie restaurants to their offerings and orders.

II.III.IV Purpose of Customer Table

This table stores details about customers who use the "Lion Bites" system to place orders. The customerID serves as a unique identifier for each customer, while fields like firstname (varchar(20)), lastname (varchar(20)), contactnumber (varchar(11)), email (varchar(50)), district (varchar(30)), and city (varchar(30)) capture their personal and location information. The table’s purpose is to maintain customer profiles,track their order history, and manage interactions with them. It links to the Orders table via the customerID foreign key, associating each customer with their respective orders.

II.III.V Purpose of Orders Table

This table stores information about orders placed by customers in the "Lion Bites" system. The orderID uniquely identifies each order, and fields such as customerID (foreign key), paymentmethodID (foreign key), restaurantID (foreign key), productID (foreign key), quantity (int), amount (double), mode (varchar(30)), orderDate (date), and orderTime (time) detail the customer, payment method, restaurant, products ordered, quantity, total cost, payment mode, and order timing. Its purpose is to track and manage order transactions, connecting to the Customer, PaymentMethod, Restaurant, and

Product tables through foreign keys to provide a complete picture of each order.

II.III.VI Purpose of Product Table

This table stores information about the products available in the "Lion Bites" system. The productID uniquely identifies each product, while pname (varchar(50)), restaurantID (foreign key), and price (int) specify the product name, the restaurant offering it, and its price. The table's purpose is to maintain a catalog of products offered by restaurants, linking to the Restaurant table via the restaurantID foreign key and to the Orders table via the productID foreign key. This allows tracking of product availability, pricing, and their use in orders.

II.IV Relations

From Table	To Table	Relation
Customer	PaymentMethod	A customer can **choose** a one-to-many payment method.
Customer	Restaurant	A customer can **choose** a zero-to-many restaurant.
Customer	Product	A customer can **choose** a zero-to-many product.
Customer	Orders	A customer can **place** zero-to-many orders.
Customer	Category	A customer can **choose** a zero-to-many category.
Orders	Customer	An order can be **placed** by one-to-one customer.
Orders	PaymentMethod	An order can **use** a one-to-one payment

		method.
Orders	Restaurant	An order **has** a one-to-one restaurant.
Orders	Product	An order **has** one-to-many product.
Restaurant	Customer	A restaurant can be **chosen** by zero-to-many customers.
Restaurant	Product	A restaurant **offers** one-to-many products.
Restaurant	Category	A restaurant can be **contained** by one-to-one category
Restaurant	Orders	A restaurant **has** zero-to-many orders.
Product	Customer	A product can be **chosen** by zero-to-many customers.
Product	Restaurant	A product can be **offered** by a one-to-one restaurant.
Product	Orders	A product **has** zero-to-many orders.
Category	Customer	A category can be **chosen** by zero-to-many customers.
Category	Restaurant	A category **contains** one-to-many restaurant.
PaymentMethod	Customer	A payment method can be **chosen** by zero-to-many customers.
PaymentMethod	Orders	A payment method can be **used** in zero-to-many orders.

III. References

Notion – The all-in-one workspace for your notes, tasks, wikis, and databases. (n.d.). Notion. https://believed-bongo-319.notion.site/CTINFMGL-Project-Specifications-19296450aad180aea9ebf27987415f4c