Part B: Design Normalized Database

Krishnappa, Kushal

Spring 2025

1. Introduction

This document presents a **normalized relational database schema** for managing restaurant visits. The schema follows **Third Normal Form (3NF)** to eliminate redundancy and ensure data integrity. The steps involved include listing the initial **functional dependencies** for a restaurant visit relation represented by all of the columns, identifying **entities** and their relationships, performing **decomposition** on identified **entities** and **functional dependencies** to ensure normalization to 3NF, and designing an **Entity-Relationship Diagram (ERD)** to illustrate the final structure.

2. Functional Dependencies Identified For CSV Data

Initial data for the restaurant visits is a single relation represented by all of the columns. The functional dependencies are identified from the raw data to understand the relationships between the attributes.

$$\text{VisitID} \rightarrow \left\{ \begin{array}{l} \text{Restaurant, ServerEmpID, ServerName, StartDateHired, EndDateHired, HourlyRate,} \\ \text{ServerBirthDate, ServerTIN, VisitDate, VisitTime, MealType, PartySize, Genders,} \\ \text{WaitTime, CustomerName, CustomerPhone, CustomerEmail, LoyaltyMember,} \\ \text{FoodBill, TipAmount, DiscountApplied, PaymentMethod, OrderedAlcohol, AlcoholBill} \end{array} \right\}$$

$$\text{ServerEmpID} \rightarrow \{\ \text{ServerName, StartDateHired, EndDateHired, HourlyRate, ServerBirthDate, ServerTIN}\ \}$$

$$\text{CustomerName} \rightarrow \{\text{CustomerPhone, CustomerEmail, LoyaltyMember}\}$$

3. Normalization

3.1 Entities Identified and Attributes Grouped

1. Visit

- **key**: `VisitID`
- `Restaurant`
- `VisitDate`
- `VisitTime`
- `MealType`
- `PartySize`
- `Genders`
- `WaitTime`

2. **Server**

- **key**: `ServerEmpID`
- `ServerName`
- `StartDateHired`
- `EndDateHired`
- `HourlyRate`
- `ServerBirthDate`
- `ServerTIN`

3. **Customer**

- **key**: `CustomerID`
- `CustomerPhone`
- `CustomerName`
- `CustomerEmail`
- `LoyaltyMember`

4. **Billing**

- **key**: `BillingID`
- `FoodBill`
- `TipAmount`
- `DiscountApplied`
- `PaymentMethod`
- `orderedAlcohol`
- `AlcoholBill`

Relationship Between Identified Entities

1. **Visit - Customer Relationship (One-to-Many)**

- **Explanation**: A **customer** can make multiple **visits**, but each **visit** is associated with a single **customer**.
- **Cardinality**: **1 Customer → M Visits** (1:N)

2. **Visit - Server Relationship (One-to-Many)**

- **Explanation**: A **server** can attend multiple **visits**, but each **visit** is handled by a single **server**.
- **Cardinality**: **1 Server → M Visits** (1:N)

3. **Visit - Billing Relationship (One-to-One)**

- **Explanation**: Each **visit** has exactly **one** related **billing entry**, and each **billing entry** belongs to exactly **one visit**.
- **Cardinality**: **1 Visit → 1 Billing** (1:1)

3.2 Approach To Normalization

- **1NF**: Each attribute is atomic. All the tables have separate columns for ID's which will uniquely identify each row.
- **2NF**: Partial dependencies are eliminated by decomposing the relation into multiple relations and ensuring each relation has a single primary key.
- **3NF**: Transitive dependencies are removed by creating separate relations for the dependent attributes.

Functional Dependencies From Normalized Database

$$\text{PaymentID} \rightarrow \{\text{PaymentMethod}\}$$

$$\text{CustomerID} \rightarrow \{\text{CustomerName}, \text{CustomerPhone}, \text{CustomerEmail}, \text{LoyaltyMember}\}$$

$$\text{RestaurantID} \rightarrow \{\text{RestaurantName}\}$$

$$\text{MealTypeID} \rightarrow \{\text{MealType}\}$$

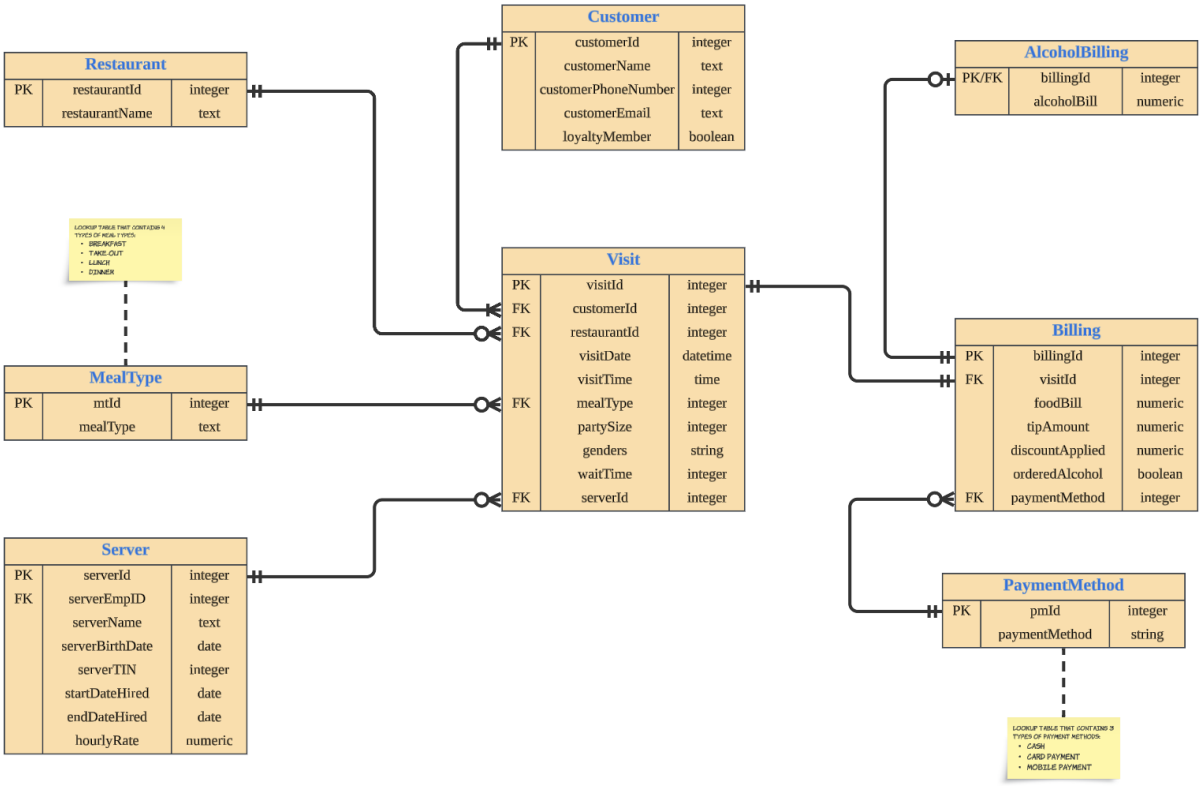
$$\text{ServerEmpID} \rightarrow \{\ \text{ServerName}, \text{StartDateHired}, \text{EndDateHired}, \text{HourlyRate}, \text{ServerBirthDate}, \text{ServerTIN}\ \}$$

$$\text{VisitID} \rightarrow \left\{ \begin{array}{l} \text{RestaurantID}, \text{ServerEmpID}, \text{VisitDate}, \text{VisitTime}, \text{MealTypeID}, \text{PartySize}, \text{Genders}, \\ \text{WaitTime}, \text{CustomerID} \end{array} \right\}$$

$$\text{BillID} \rightarrow \left\{ \begin{array}{l} \text{VisitID}, \text{FoodBill}, \text{TipAmount}, \text{DiscountApplied}, \text{PaymentID}, \text{OrderedAlcohol}, \\ \text{AlcoholBill} \end{array} \right\}$$

4. ERD for Normalized Tables

Entity-Relationship Diagram (ERD) in Crow's Foot Notation


Customer
PK customerId integer
customerName text
customerPhoneNumber integer
customerEmail text
loyaltyMember boolean
Restaurant
PK restaurantId integer
restaurantName text
AlcoholBilling
PK/FK billingId integer
alcoholBill numeric
LOOKUP TABLE THAT CONTAINS 4 TYPES OF MEALTIMES
• BREAKFAST
• TAKE-OUT
• LUNCH
• DINNER
Visit
PK visitId integer
FK customerId integer
FK restaurantId integer
visitDate datetime
visitTime time
FK mealType integer
partySize integer
genders string
waitTime integer
FK serverId integer
Billing
PK billingId integer
FK visitId integer
foodBill numeric
tipAmount numeric
discountApplied numeric
orderedAlcohol boolean
FK paymentMethod integer
MealType
PK mtId integer
mealType text
Server
PK serverId integer
FK serverEmpID integer
serverName text
serverBirthDate date
serverTIN integer
startDateHired date
endDateHired date
hourlyRate numeric
PaymentMethod
PK pmId integer
paymentMethod string
LOOKUP TABLE THAT CONTAINS 3 TYPES OF PAYMENT METHODS
• CASH
• CARD PAYMENT
• MOBILE PAYMENT