

Project Phase #1

Team sqLITE:

Sarthak Agrawal - 2019115003

Ainesh Sannidhi - 2019101067

Aadarsh Raghunathan - 2019113021

Introduction

A general database for feedback forms for restaurant chains to store the surveys of their customers over all their branches. The mini-world consists of a restaurant chain, its branches, employees (namely the managers, waiters and chefs), the dishes served and the feedback obtained from the customers.

The users of this database would be the restaurant chain's upper-management. The database system can be utilised to evaluate the performance of branches, dishes and employees. For example, to see which employees are underperforming, or which dishes are getting good reviews. With the data obtained, users can address the issues evident from the data trends, such as removing or changing a dish if its average rating is low.

This database can also be generalised to be used by online food delivery platforms like Foodpanda, Swiggy and Zomato. Such companies can rank restaurants and their food with detail for customers, and it is increasingly common to find two branches of an outlet in close vicinity. For such companies, the entities will change slightly, but the usability is not affected in any way whatsoever. This showcases the scale at which the database system operates, its flexibility and the degree of generalisation that allows this adaptability.

Database Requirements

1. Entities (at least 5, with at least 2 weak entities)
 - a. Strong Entities:
 - Employee(superclass of chef, manager, and waiter)
 - Employee_id(key)
 - Chef
 - Employee_id(key), position
 - Waiter
 - Employee_id(key),
 - Restaurant
 - Cin_num(key), name
 - Manager
 - Employee_id(key),
 - Recipe:
 - Recipe_name(key), Ingredients(multi-valued, string)

- Customer:
 - Customer_id(key)[composite attribute with phone number and time]

b) Weak Entities:

- Branch
 - Branch_ID
 - Identifying Relationship: Branch_of
- Dish
 - Price, Meal (The meal of the day when the dish is served. Domain-Breakfast, Lunch, Snack, Dinner)
 - Identifying relationship: From

2. Relationships between Entities:-

- Feedback:
 - Degree: 4-ary
 - Participating entities: Dish, Chef, Waiter, Customer
 - Description: A customer 'c' gives feedback for a dish 'd' prepared by chef 'C' and served by waiter 'w'
 - Participation: All entities have total participation in the relationship.
 - Attributes: Rating(integer from 1-10), Suggestion(String)
- Branch_of: 1:N binary identifying relationship between strong entity restaurant and weak entity branch (both total participation).
- Employ: 1:N between branch and employee (both total participation).
- From: Identifying binary relationship between strong entity recipe and weak entity dish. The cardinality ratio for this relationship is 1:1 (total participation from both entities)
- (BONUS)Supervisor: Recursive relation of Employee. Each employee has a distinct role of either manager or subordinate. A manager has at least one subordinate and every subordinate has exactly one manager.

3. Entity with 2 key attributes:- Recipe

2 weak-entity types:-

- Branch
- Dish

- 4-degree relationship:- Feedback (between customer, waiter, dish and chef)
- Subclasses:- Employee superclass has Waiter, Chef and Manager as subclasses
- Composite- Customer_id; Multivalued- Meal; Derived- Position, Name

Functional Requirements

I. Retrieval:

(A) Query functions

1. Selection
 - a. Feedback
 - i. A particular Feedback
 - ii. All Feedbacks for a specific employee, a dish, or a branch
 - b. Supervisor of a specific employee
 - c. Subordinates for a specific supervisor
 - d. Ingredients for a particular dish
2. Projection
 - a. Feedback for all dishes of a meal-type 'x' ('x' can be breakfast, lunch, etc.)
3. Aggregate
 - a. Average rating across all branches
 - b. Average/Max/Min rating for a branch/a dish/an employee
 - c. Average subordinate rating for a supervisor
 - d. Number of employees under a rating 'x' across all branches
4. Search
 - a. Search in the suggestion attribute of feedback relation

(B) Analysis:

1. All branches/dishes/employees with an average rating greater/less than a value 'x'
2. All employees/dishes with an average rating greater/less than 'x' for a branch 'y'
3. All supervisors with an average subordinate rating greater/less than 'x' for a particular branch

II. Modification

1. Insertion
 - a. Feedback (with rating and suggestion)
 - b. Employee/Dish/Branch/Restaurant/Recipe
2. Update
 - a. Price of a Dish
3. Delete
 - a. Employee/Dish/Branch/Recipe

Tentative ER diagram:

