Delivery Management Database Design Report

Database Systems Course - 02360363

Student: Yara Zoabi ID: 212264048

Student: Diar Batheesh ID: 213859325

Date: June 7, 2025

Contents

1 Introduction

This document provides a comprehensive description of the relational database designed to support the operations of a delivery–oriented restaurant. The following pages present the rationale behind every table, constraint, and derived view.

2 Requirements and Assumptions

Before designing the schema, the following functional and non-functional requirements were identified:

- **Customer management**: store personal details, enforce an age limit, and guarantee unique identifiers.
- **Orders**: track delivery address, dispatch date, and a per-order delivery fee.
- **Menu items (dishes)**: maintain current price and availability (*is_active*). Historical prices are captured implicitly through `DishOrders.price`.
- **Many-to-many associations**: an order may contain multiple dishes and each dish can appear in many orders.
- **Ratings**: customers may rate any dish once, on a scale from 1 to 5.
- **Business intelligence**: compute metrics such as order totals, best-rated dishes, similar customers, and monthly profits.
- **Data integrity**: all foreign keys must cascade on delete; attempts to insert invalid data should fail fast using `CHECK` constraints.

3 Conceptual Model

Key entities and relationships are:

Entity	*Attributes*
Customer	cust_id (PK), full_name, age, phone
Order	order_id (PK), date, delivery_fee, delivery_address
Dish	dish_id (PK), name, price, is_active

Relationships:

- **CustomerOrders** (1:N): each customer may place multiple orders, but each order belongs to exactly one customer.
- **DishOrders** (M:N): captures the quantity and effective price of each dish within an order.
- **Ratings** (M:N): a weak relationship between customers and dishes with additional attribute *rating*.

4 Relational Schema

4.1 Tables

Table	**Definition (Data Definition Languag excerpt)**
Customers	`CREATE TABLE Customers( cust_id INTEGER PRIMARY KEY CHECK( cust_id>0), full_name TEXT NOT NULL, age INTEGER NOT NULL CHECK(age BETWEEN 18 AND 120), phone TEXT NOT NULL CHECK( LENGTH(phone)=10));`
Orders	`CREATE TABLE Orders( order_id INTEGER PRIMARY KEY CHECK( order_id>0), date TIMESTAMP NOT NULL, delivery_fee DECIMAL NOT NULL CHECK(delivery_fee>=0), delivery_address TEXT NOT NULL CHECK(LENGTH(delivery_address)>=5));`
Dishes	`CREATE TABLE Dishes( dish_id INTEGER PRIMARY KEY CHECK( dish_id>0), name TEXT NOT NULL CHECK(LENGTH(name)>=4), price DECIMAL NOT NULL CHECK(price>0), is_active BOOLEAN NOT NULL );`
CustomerOrders	`CREATE TABLE CustomerOrders( order_id INTEGER PRIMARY KEY , cust_id INTEGER REFERENCES Customers(cust_id)ON DELETE CASCADE, FOREIGN KEY(order_id)REFERENCES Orders(order_id )ON DELETE CASCADE );` The `PRIMARY KEY` guarantees the one-to-one relationship assumption: each order belongs to exactly one customer.
DishOrders	`CREATE TABLE DishOrders( order_id INTEGER REFERENCES Orders (order_id)ON DELETE CASCADE, dish_id INTEGER REFERENCES Dishes(dish_id)ON DELETE CASCADE, amount INTEGER NOT NULL CHECK(amount >= 0), price DECIMAL NOT NULL, PRIMARY KEY (order_id, dish_id));` Historical pricing is preserved via `price`.
Ratings	`CREATE TABLE Ratings( cust_id INTEGER REFERENCES Customers (cust_id)ON DELETE CASCADE, dish_id INTEGER REFERENCES Dishes(dish_id)ON DELETE CASCADE, rating INTEGER NOT NULL CHECK(rating BETWEEN 1 AND 5), PRIMARY KEY(cust_id, dish_id ));` Each customer may rate a dish at most once.

4.2 Integrity and Quality Constraints

- **Domain checks**: e.g. `age BETWEEN 18 AND 120`, `delivery_fee >= 0`. These prevent illogical values from entering the database.
- **Referential actions**: `ON DELETE CASCADE` ensures orphan records are never left behind—if a dish is deleted, its ratings vanish automatically.

- **Deletion Semantics:** Our database design carefully follows the specified deletion requirements. When a **customer** is deleted, all their related data—including ratings and order ownership links—is automatically removed using `ON DELETE CASCADE`. However, the actual **orders** themselves remain intact, allowing the restaurant to preserve historical profit records even after the customer no longer exists. Similarly, when a **dish** is deleted, all dependent entries such as dish orders and customer ratings are also removed, ensuring that no orphaned or inconsistent data remains. This behavior is achieved through referential actions and foreign key constraints, which enforce clean and complete deletions in line with system requirements.

5 Derived Structures (Views)

Views encapsulate recurring analytical queries, improving code reuse and readability.

5.1 total_price_per_order

Computes the grand total (dishes + delivery fee) for every order.

```
SELECT O.order_id,
       (COALESCE(SUM(D.amount * D.price), 0) + O.delivery_fee) AS
           total_price
FROM DishOrders D RIGHT OUTER JOIN Orders O USING(order_id)
GROUP BY O.order_id, O.delivery_fee;
```

A `RIGHT JOIN` ensures orders without any dishes still appear with a `0` total.

5.2 sort_ratings_desc

Maintains the current top 5 dishes by average rating (default 3 for unrated dishes). This aids menu curation and personalized recommendations.

5.3 compared_prices

For each historical selling price of a dish, calculates the average revenue generated *per order line*. It feeds the advanced query `get_non_worth_price_increase`.

5.4 similar_customers

A recursive CTE implements collaborative filtering: two customers are related if they both gave positive ratings (>3) to at least one common dish, and this relation is propagated transitively. The result helps recommend dishes a customer has not yet ordered.

5.5 monthly_orders and monthly_profit

Aggregate revenue per order and roll it up to calendar months. These are materialized as views (rather than tables) because the underlying data changes frequently and the expected query latency is low.

6 Front-End Demonstration

To illustrate the database in action, a lightweight web interface called *Yummify* was developed (using Streamlit). The screenshots below show real data flowing through the API and the analytical views described earlier.

1. **Dashboard Overview** — Customers and Orders tables rendered immediately after initialisation. Note how age and delivery-fee constraints are respected (Figure 1).

2. **Menu Management** — Active dishes alongside their historical prices inside `DishOrders` and the master `Dishes` catalog. The checkbox column reflects the `is_active` flag. (Figure 2)

3. **Order–Customer Mapping** — The `Customers Orders` view shows which customer placed each order, mapping each order to one customer, while each customer may place multiple orders. (Figure 3).

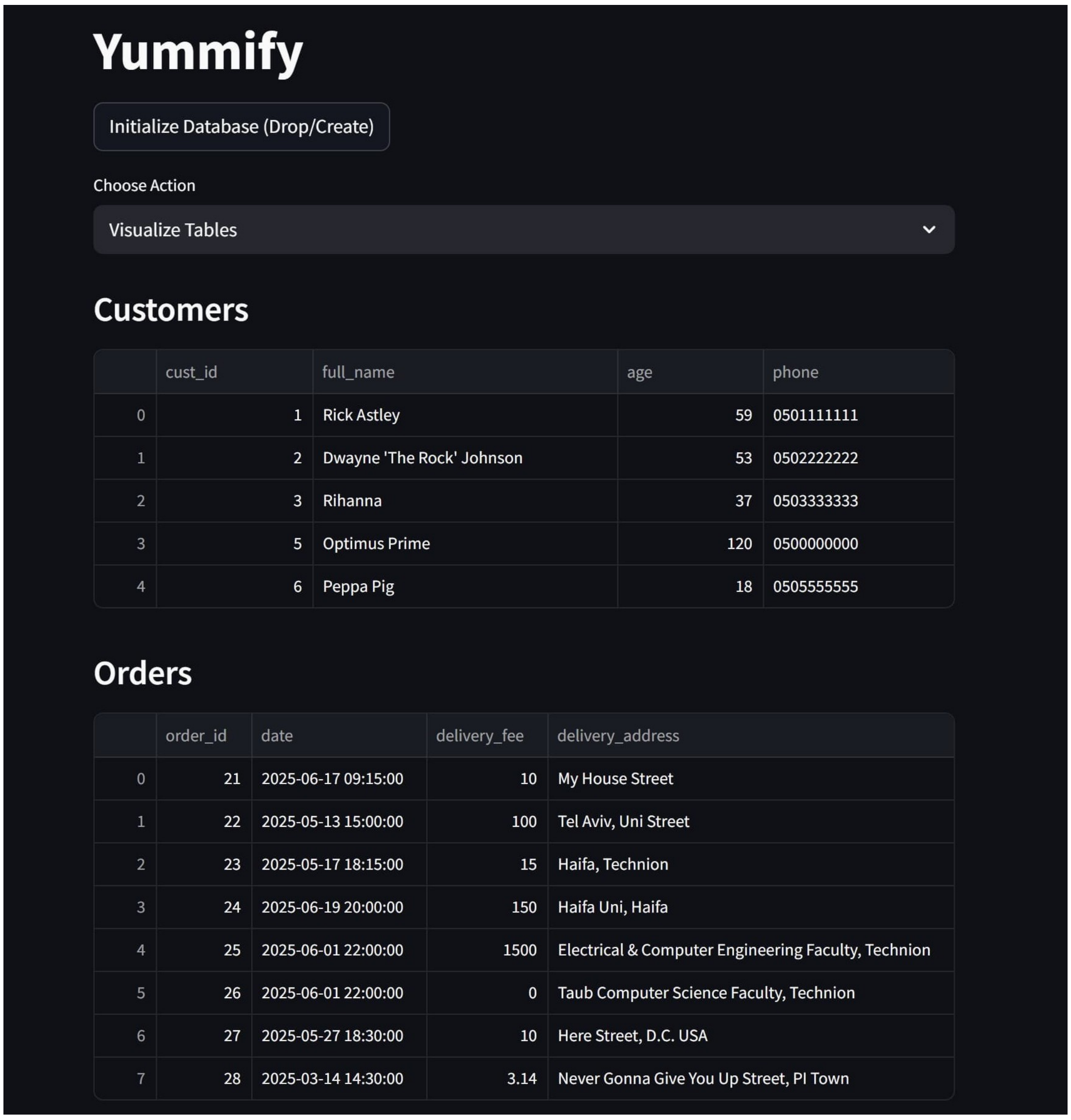
4. **Analytics Widgets** — Total price of every order (Figure 4).

5. **Maximum Average Spending** — Identifies customer(s) with the highest average total order value across all their orders. Here, Optimus Prime emerges as the top spender (Figure 5).

6. **Order 26 Details** — For order ID 26, the "Dishes ordered" view displays a bulk purchase of 10 pizzas and 1 banana, calculates the "Order Total" of $1005.00, and shows the corresponding customer information for Optimus Prime (Figure 6).

7. **Order 27 Details** — For order ID 27, the "Dishes ordered" view lists 3 bananas and 3 falafel, computes the "Order Total" of $115.00, and displays the customer details for Dwayne 'The Rock' Johnson (Figure 7).

8. **Order 28 Details** — For order ID 28, the view presents three items including Wagyu steak, two bananas at fractional pricing, and grilled cheese, resulting in an "Order Total" of $352.14 for Rick Astley (Figure 8).



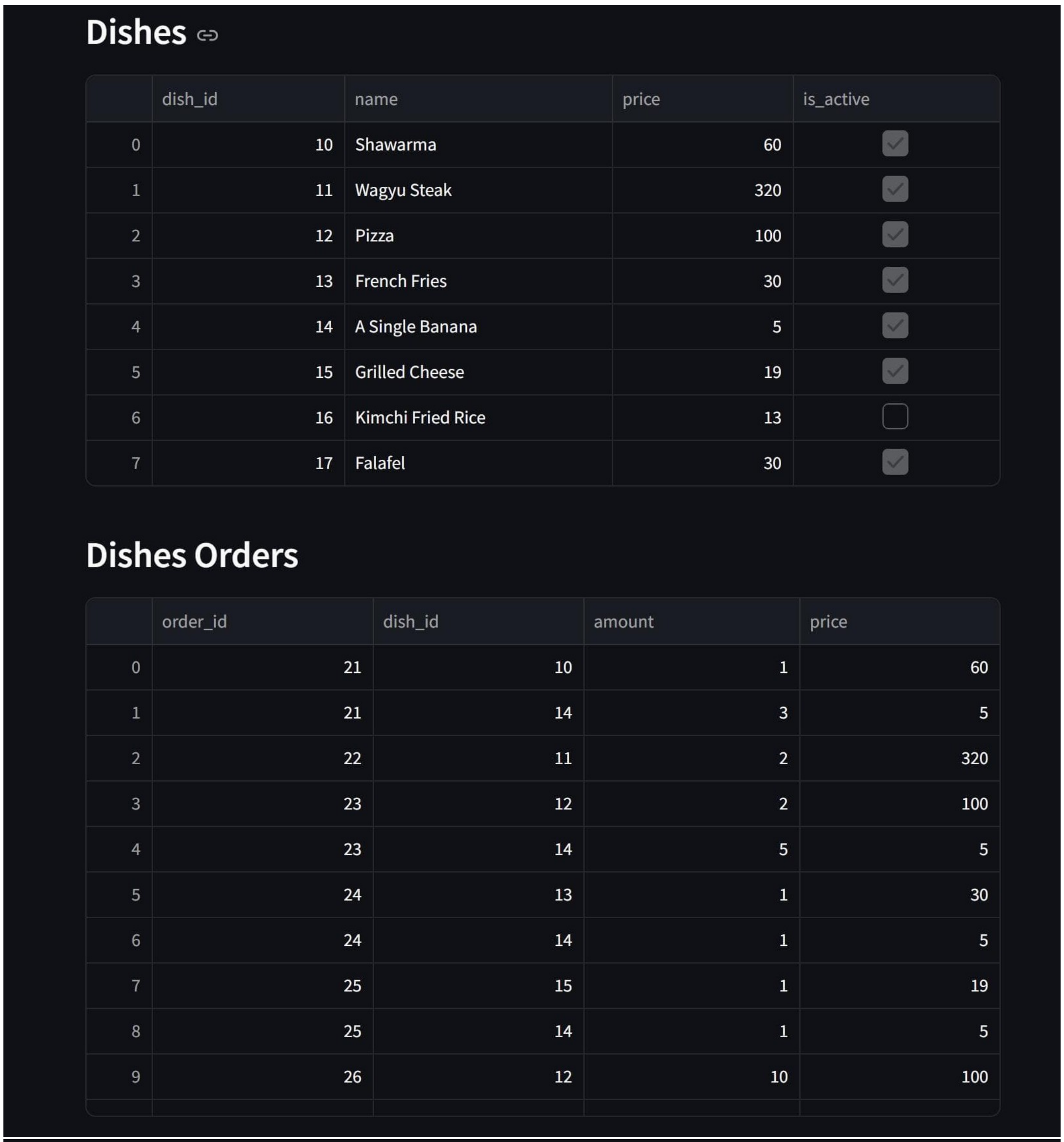
Customers

	cust_id	full_name	age	phone
0	1	Rick Astley	59	0501111111
1	2	Dwayne 'The Rock' Johnson	53	0502222222
2	3	Rihanna	37	0503333333
3	5	Optimus Prime	120	0500000000
4	6	Peppa Pig	18	0505555555

Orders

	order_id	date	delivery_fee	delivery_address
0	21	2025-06-17 09:15:00	10	My House Street
1	22	2025-05-13 15:00:00	100	Tel Aviv, Uni Street
2	23	2025-05-17 18:15:00	15	Haifa, Technion
3	24	2025-06-19 20:00:00	150	Haifa Uni, Haifa
4	25	2025-06-01 22:00:00	1500	Electrical & Computer Engineering Faculty, Technion
5	26	2025-06-01 22:00:00	0	Taub Computer Science Faculty, Technion
6	27	2025-05-27 18:30:00	10	Here Street, D.C. USA
7	28	2025-03-14 14:30:00	3.14	Never Gonna Give You Up Street, PI Town

Figure 1: Initial dashboard with Customers and Orders tables displayed.

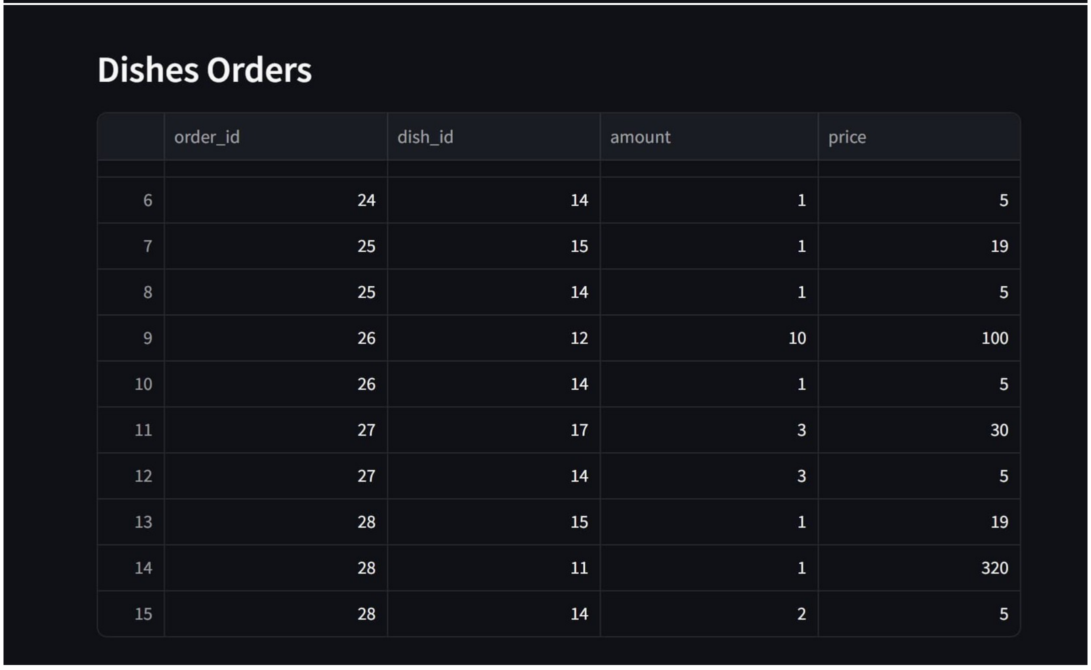


Dishes

	dish_id	name	price	is_active
0	10	Shawarma	60	✓
1	11	Wagyu Steak	320	✓
2	12	Pizza	100	✓
3	13	French Fries	30	✓
4	14	A Single Banana	5	✓
5	15	Grilled Cheese	19	✓
6	16	Kimchi Fried Rice	13	☐
7	17	Falafel	30	✓

Dishes Orders

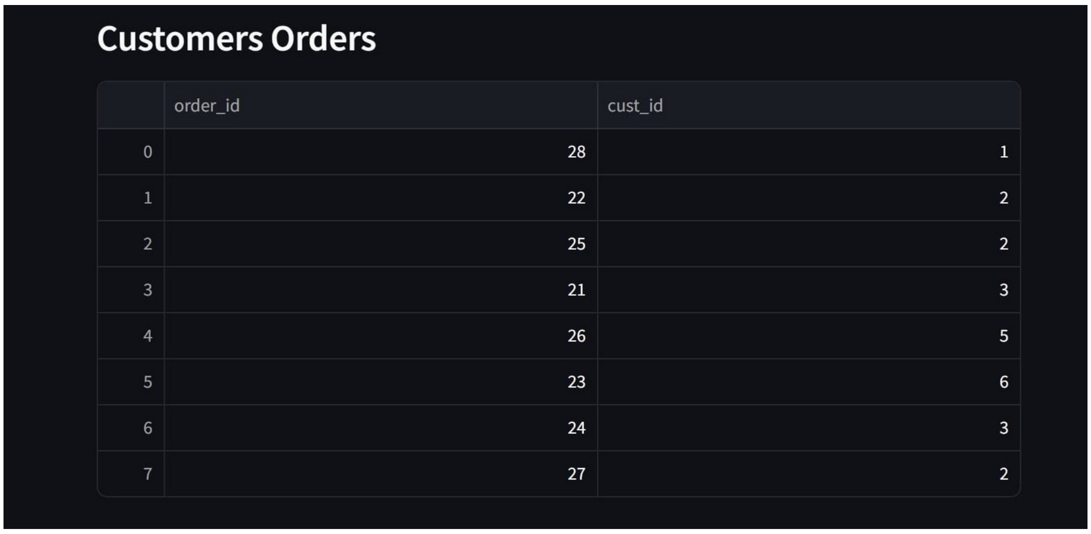
	order_id	dish_id	amount	price
0	21	10	1	60
1	21	14	3	5
2	22	11	2	320
3	23	12	2	100
4	23	14	5	5
5	24	13	1	30
6	24	14	1	5
7	25	15	1	19
8	25	14	1	5
9	26	12	10	100



Dishes Orders

	order_id	dish_id	amount	price
6	24	14	1	5
7	25	15	1	19
8	25	14	1	5
9	26	12	10	100
10	26	14	1	5
11	27	17	3	30
12	27	14	3	5
13	28	15	1	19
14	28	11	1	320
15	28	14	2	5

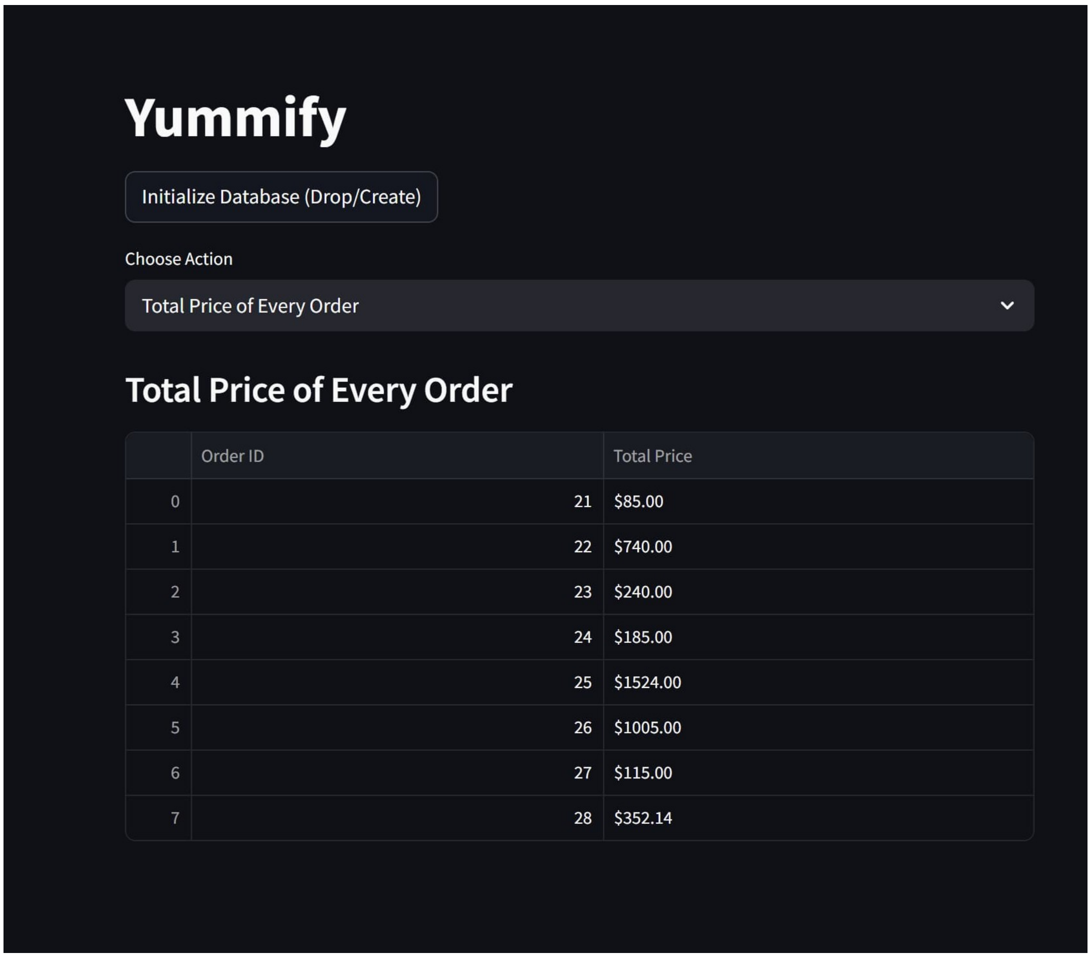
Figure 2: Active dishes and their availability, along with associated orders and prices.



Customers Orders

	order_id	cust_id
0	28	1
1	22	2
2	25	2
3	21	3
4	26	5
5	23	6
6	24	3
7	27	2

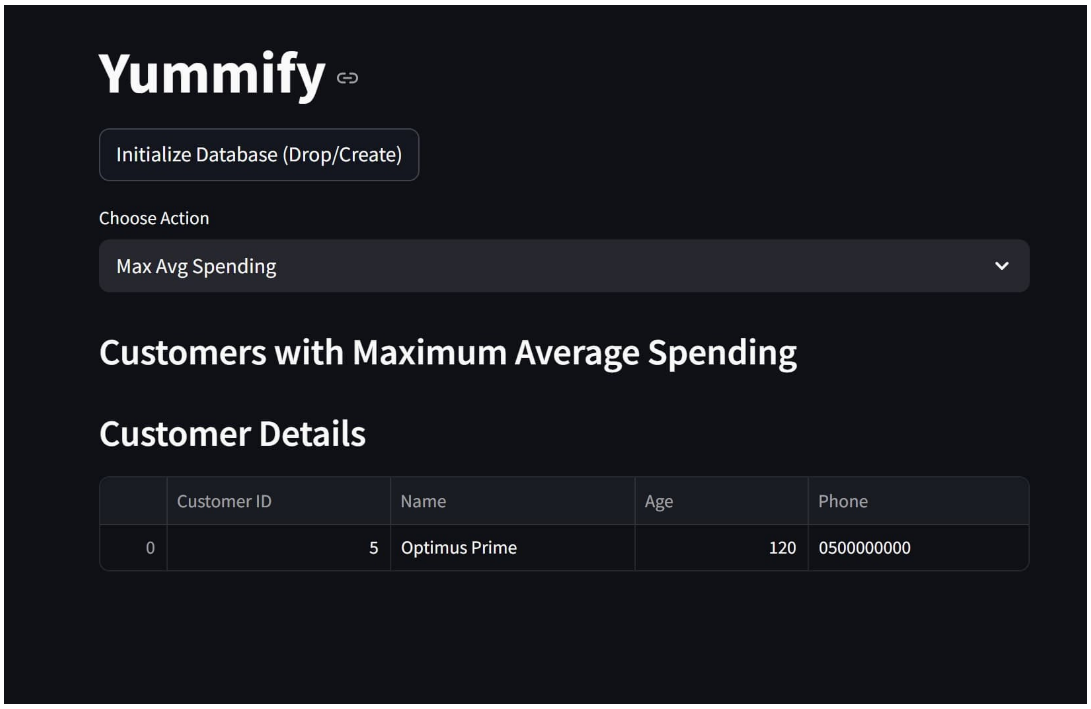
Figure 3: Customers Orders view mapping an order to a single customer.



Total Price of Every Order

	Order ID	Total Price
0	21	$85.00
1	22	$740.00
2	23	$240.00
3	24	$185.00
4	25	$1524.00
5	26	$1005.00
6	27	$115.00
7	28	$352.14

Figure 4: Total price of every order.

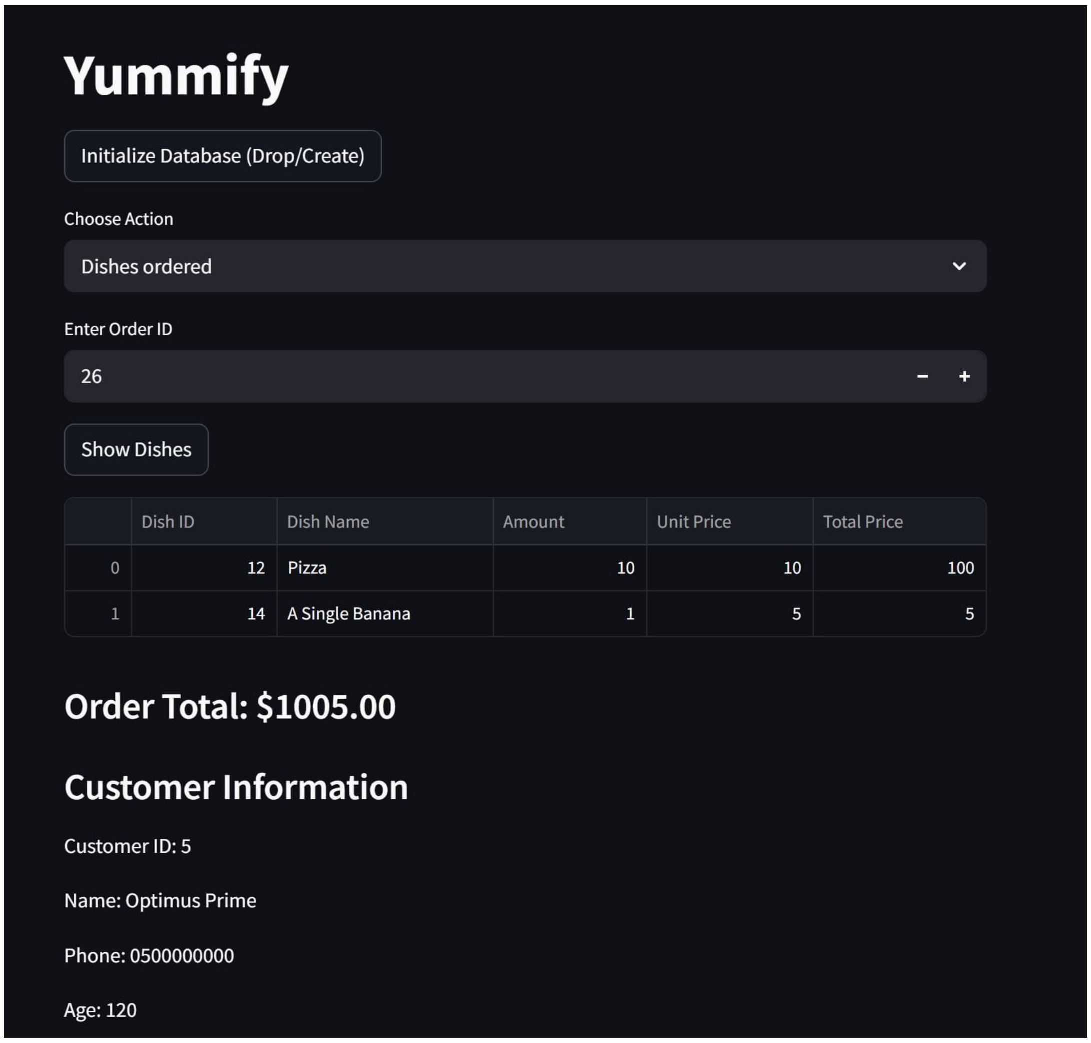


Customers with Maximum Average Spending

Customer Details

	Customer ID	Name	Age	Phone
0	5	Optimus Prime	120	0500000000

Figure 5: Customer(s) with the highest average spending across all orders.



Choose Action

Dishes ordered

Enter Order ID

26

	Dish ID	Dish Name	Amount	Unit Price	Total Price
0	12	Pizza	10	10	100
1	14	A Single Banana	1	5	5

Order Total: $1005.00

Customer Information

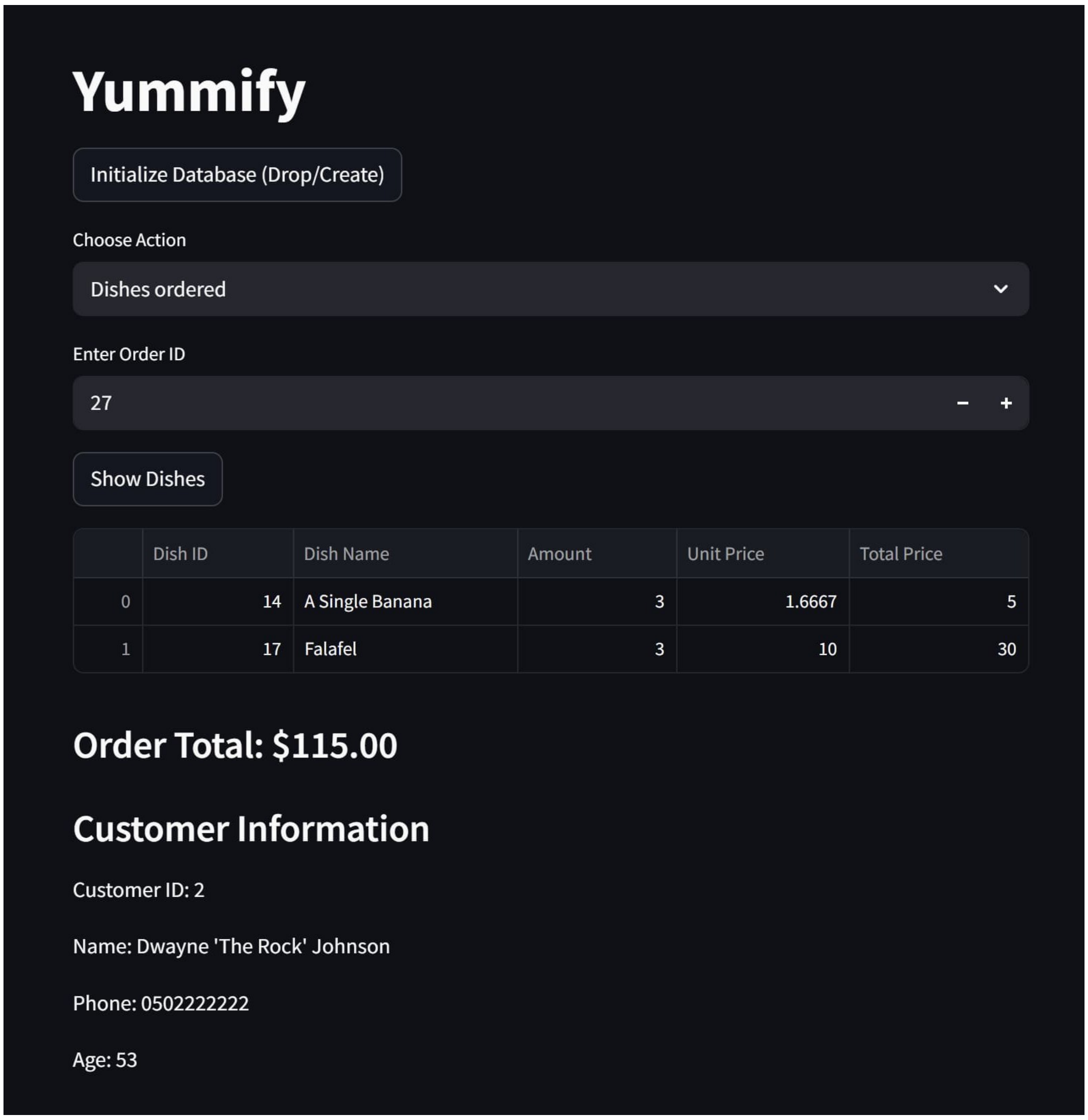
Customer ID: 5

Name: Optimus Prime

Phone: 0500000000

Age: 120

Figure 6: Order 26 detail view: dishes ordered and total price with customer info.



	Dish ID	Dish Name	Amount	Unit Price	Total Price
0	14	A Single Banana	3	1.6667	5
1	17	Falafel	3	10	30

Order Total: $115.00

Customer Information

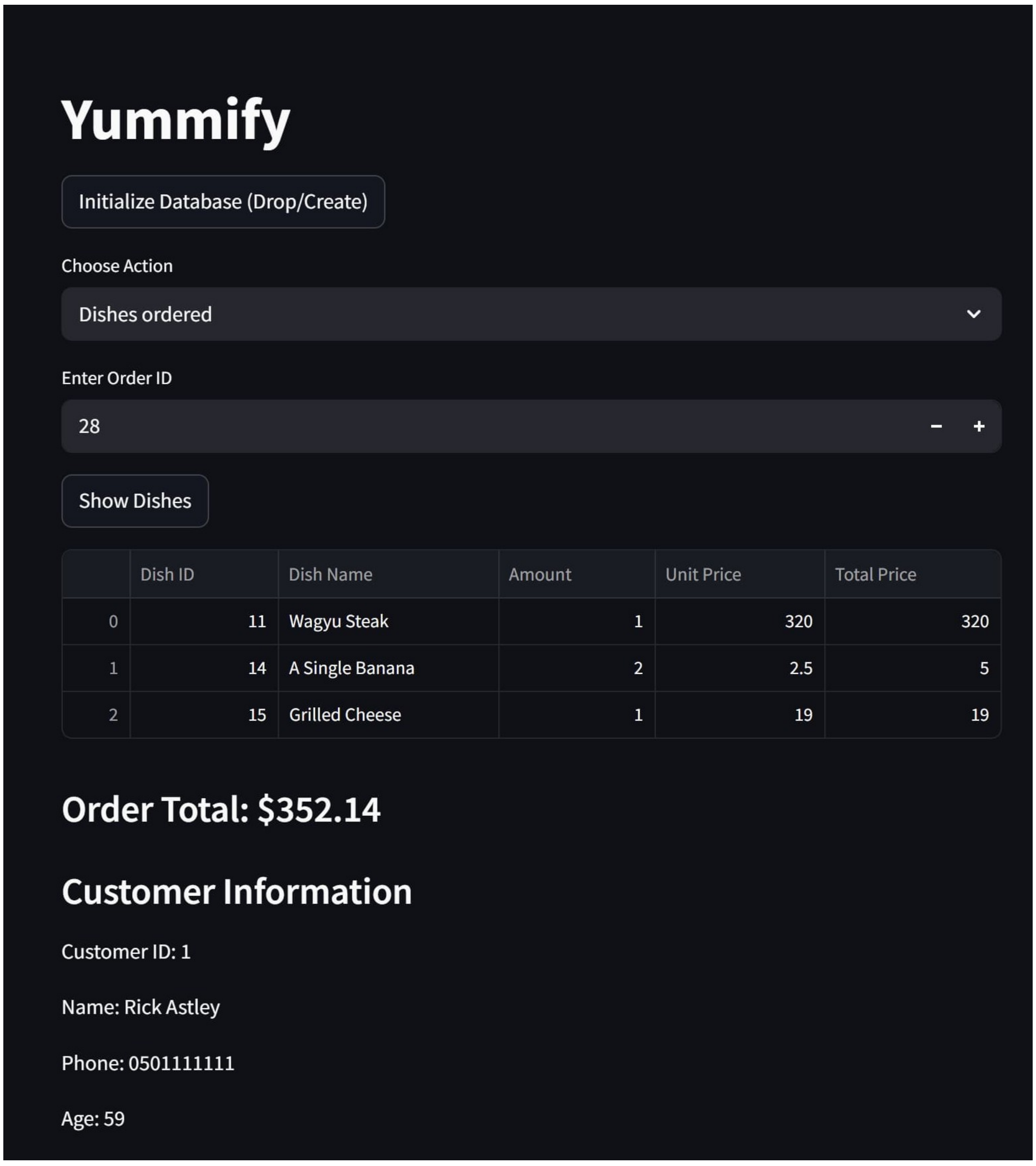
Customer ID: 2

Name: Dwayne 'The Rock' Johnson

Phone: 0502222222

Age: 53

Figure 7: Order 27 detail view: dishes ordered and total price with customer info.



	Dish ID	Dish Name	Amount	Unit Price	Total Price
0	11	Wagyu Steak	1	320	320
1	14	A Single Banana	2	2.5	5
2	15	Grilled Cheese	1	19	19

Order Total: $352.14

Customer Information

Customer ID: 1

Name: Rick Astley

Phone: 0501111111

Age: 59

Figure 8: Order 28 detail view: dishes ordered and total price with customer info.