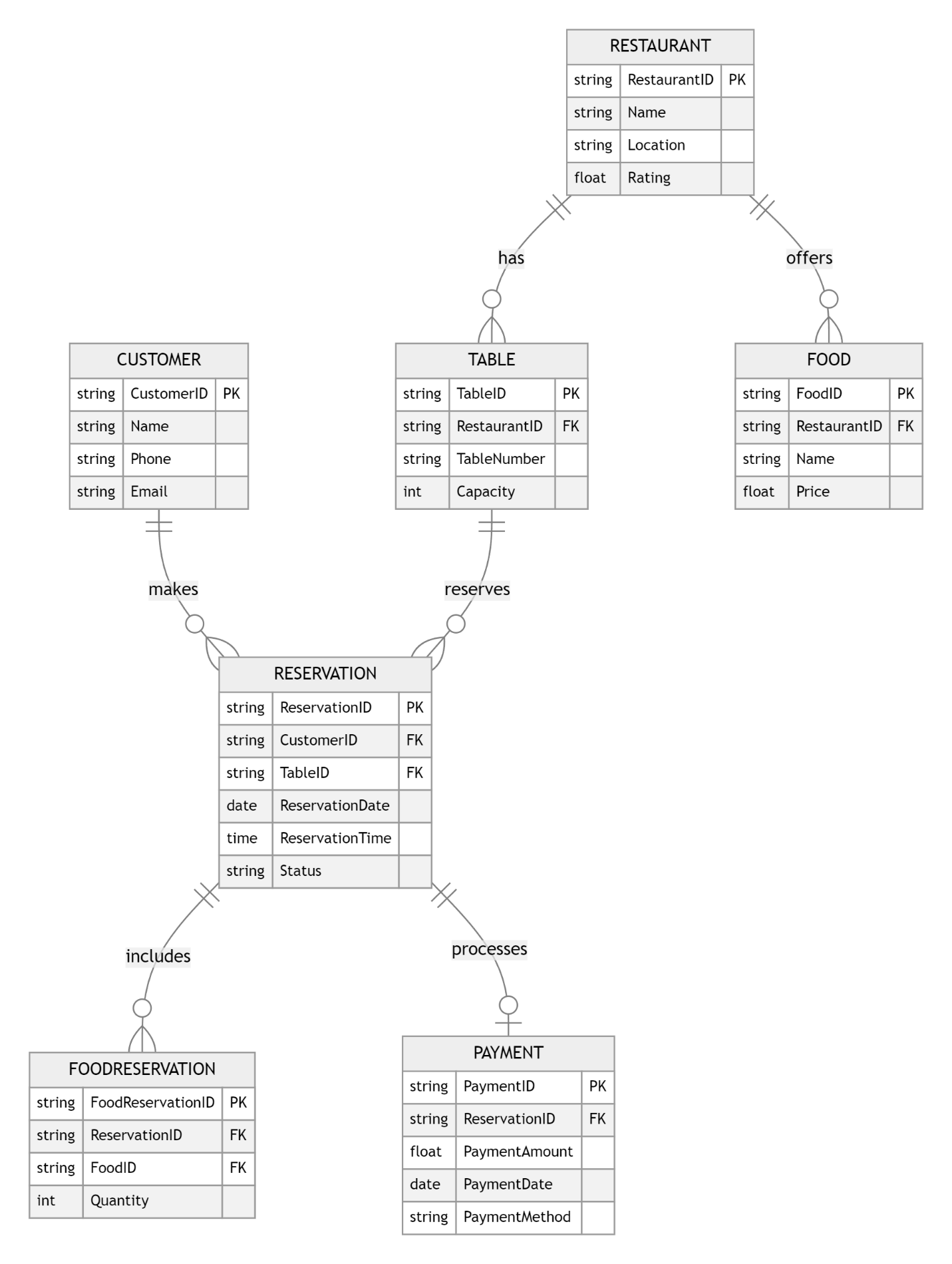
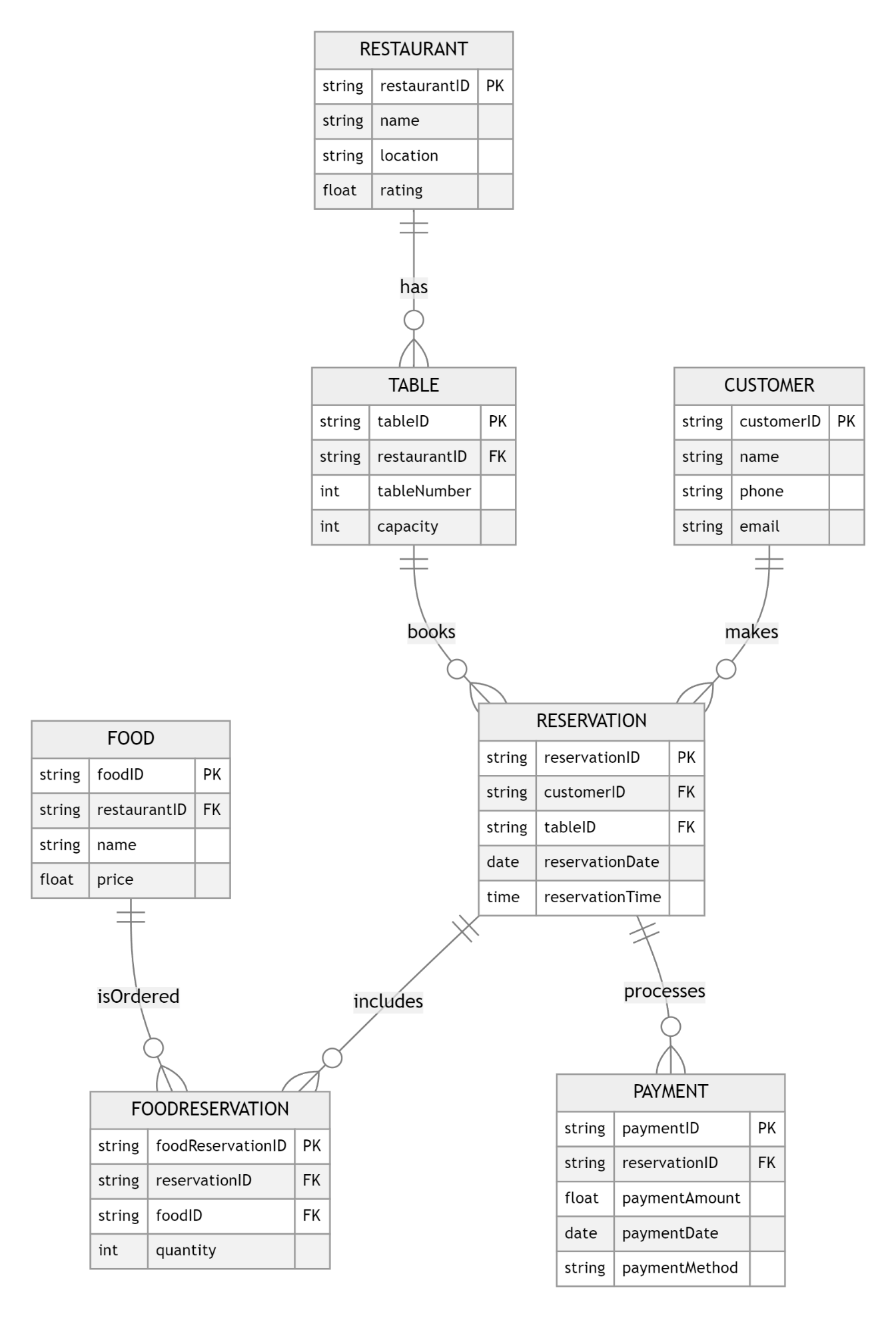
ERD of Restaurant System



Here is the **updated Entity-Relationship Diagram (ERD)** for the restaurant reservation system, including the functionality for reserving, canceling, and processing payments.

### Updated Entities and Attributes

1. **Customer**
   * **CustomerID** (Primary Key)
   * **Name** (Name)
   * **Phone** (Phone Number)
   * **Email** (Email)
2. **Restaurant**
   * **RestaurantID** (Primary Key)
   * **Name** (Restaurant Name)
   * **Location** (Restaurant Location)
   * **Rating** (Restaurant Rating)
3. **Table**
   * **TableID** (Primary Key)
   * **RestaurantID** (Foreign Key referencing Restaurant)
   * **TableNumber** (Table Number)
   * **Capacity** (Table Capacity)
4. **Reservation**
   * **ReservationID** (Primary Key)
   * **CustomerID** (Foreign Key referencing Customer)
   * **TableID** (Foreign Key referencing Table)
   * **ReservationDate** (Reservation Date)
   * **ReservationTime** (Reservation Time)
   * **Status** (e.g., Confirmed, Canceled)
5. **Food**
   * **FoodID** (Primary Key)
   * **RestaurantID** (Foreign Key referencing Restaurant)
   * **Name** (Food Name)
   * **Price** (Food Price)
6. **FoodReservation**
   * **FoodReservationID** (Primary Key)
   * **ReservationID** (Foreign Key referencing Reservation)
   * **FoodID** (Foreign Key referencing Food)
   * **Quantity** (Quantity of Food)
7. **Payment**
   * **PaymentID** (Primary Key)
   * **ReservationID** (Foreign Key referencing Reservation)
   * **PaymentAmount** (Total Amount)
   * **PaymentDate** (Payment Date)
   * **PaymentMethod** (Payment Method, e.g., Credit Card, Cash)

### Updated Relationships

* **Customer ↔ Reservation**: A Customer can have multiple Reservations (One to Many). Each Reservation belongs to one Customer (Many to One).
* **Restaurant ↔ Table**: A Restaurant can have multiple Tables (One to Many). Each Table belongs to one Restaurant (Many to One).
* **Reservation ↔ Table**: Each Reservation is for one Table (Many to One).
* **Restaurant ↔ Food**: A Restaurant can offer multiple Foods (One to Many). Each Food belongs to one Restaurant (Many to One).
* **Reservation ↔ FoodReservation**: Each Reservation can include multiple Foods (One to Many). Each Food can be ordered multiple times in a Reservation (Many to One).
* **Reservation ↔ Payment**: Each Reservation can have one Payment (One to One). Each Payment is associated with one Reservation (One to One).
* **Reservation ↔ Status**: Each Reservation can be confirmed or canceled (as part of the Reservation entity).

### Updated Entities and Attributes

1. **Customer**
   * **CustomerID** (Primary Key)
   * **Name** (Name)
   * **Phone** (Phone Number)
   * **Email** (Email)
2. **Restaurant**
   * **RestaurantID** (Primary Key)
   * **Name** (Restaurant Name)
   * **Location** (Restaurant Location)
   * **Rating** (Restaurant Rating)
3. **Table**
   * **TableID** (Primary Key)
   * **RestaurantID** (Foreign Key referencing Restaurant)
   * **TableNumber** (Table Number)
   * **Capacity** (Table Capacity)
4. **Reservation**
   * **ReservationID** (Primary Key)
   * **CustomerID** (Foreign Key referencing Customer)
   * **TableID** (Foreign Key referencing Table)
   * **ReservationDate** (Reservation Date)
   * **ReservationTime** (Reservation Time)
   * **Status** (e.g., Confirmed, Canceled)
5. **Food**
   * **FoodID** (Primary Key)
   * **RestaurantID** (Foreign Key referencing Restaurant)
   * **Name** (Food Name)
   * **Price** (Food Price)
6. **FoodReservation**
   * **FoodReservationID** (Primary Key)
   * **ReservationID** (Foreign Key referencing Reservation)
   * **FoodID** (Foreign Key referencing Food)
   * **Quantity** (Quantity of Food)
7. **Payment**
   * **PaymentID** (Primary Key)
   * **ReservationID** (Foreign Key referencing Reservation)
   * **PaymentAmount** (Total Amount)
   * **PaymentDate** (Payment Date)
   * **PaymentMethod** (Payment Method, e.g., Credit Card, Cash)

### Updated Relationships

* **Customer ↔ Reservation**: A Customer can have multiple Reservations (One to Many). Each Reservation belongs to one Customer (Many to One).
* **Restaurant ↔ Table**: A Restaurant can have multiple Tables (One to Many). Each Table belongs to one Restaurant (Many to One).
* **Reservation ↔ Table**: Each Reservation is for one Table (Many to One).
* **Restaurant ↔ Food**: A Restaurant can offer multiple Foods (One to Many). Each Food belongs to one Restaurant (Many to One).
* **Reservation ↔ FoodReservation**: Each Reservation can include multiple Foods (One to Many). Each Food can be ordered multiple times in a Reservation (Many to One).
* **Reservation ↔ Payment**: Each Reservation can have one Payment (One to One). Each Payment is associated with one Reservation (One to One).
* **Reservation ↔ Status**: Each Reservation can be confirmed or canceled (as part of the Reservation entity).