Here is the comprehensive documentation for your *Final Restaurant Database*. This includes the updated Schema (including the new bridge table), the Relationship definitions, and the specific Cardinality/Participation rules required for academic or technical documentation.

1. Final Schema Overview

These are your 14 finalized entities.

A. Front of House (Customers & Orders)

Entity	Attribute	Type	Key	Notes
Customers	CustomerID	INT	PK	Unique ID
	FirstName	VARCHAR		
	LastName	VARCHAR		
	Phone	VARCHAR		
	Email	VARCHAR		
Reservations	ReservationID	INT	PK	
	CustomerID	INT	FK	Links to Customers
	TableID	INT	FK	Links to Tables
	DateTime	DATETIME		
	NumGuests	INT		
	Status	ENUM		Pending, Confirmed, etc.
Tables	TableID	INT	PK	
	TableNumber	INT		Physical label on table
	Capacity	INT		Max seats
Orders	OrderID	INT	PK	
	CustomerID	INT	FK	
	StaffID	INT	FK	Server who took order
	OrderType	ENUM		Dine-In, Takeout, Delivery
	TotalAmount	DECIMAL		Calculated total

B. Menu & Production (The Bridge)

Entity	Attribute	Type	Key	Notes
MenuCategories	CategoryID	INT	PK	e.g., Appetizers, Drinks
	Name	VARCHAR		
MenuItems	MenuItemID	INT	PK	
	CategoryID	INT	FK	
	Name	VARCHAR		
	Price	DECIMAL		Selling Price
OrderItems	OrderItemID	INT	PK	
	OrderID	INT	FK	
	MenuItemID	INT	FK	
	Quantity	INT		
RecipeIngredients	RecipeID	INT	PK	*(The Bridge Table)*
	MenuItemID	INT	FK	Which dish?
	InventoryID	INT	FK	Which ingredient?
	QuantityRequired	DECIMAL		e.g., 0.2 (kg)

C. Back of House (Inventory & Suppliers)

Entity	Attribute	Type	Key	Notes
InventoryItems	InventoryID	INT	PK	Ingredients stock
	Name	VARCHAR		
	Quantity	INT		Current stock count
	Unit	VARCHAR		kg, liters, pcs
Suppliers	SupplierID	INT	PK	
	Name	VARCHAR		
	ContactEmail	VARCHAR		
SupplyOrders	SupplyOrderID	INT	PK	Purchase Order to supplier
	SupplierID	INT	FK	
	TotalCost	DECIMAL		Cost to restaurant
SupplyOrderItems	SupplyOrderItemID	INT	PK	
	SupplyOrderID	INT	FK	
	InventoryID	INT	FK	What was bought

Entity	Attribute	Type	Key	Notes
	CostPerUnit	DECIMAL		

D. Staff

Entity	Attribute	Type	Key	Notes
Staff	StaffID	INT	PK	
	RoleID	INT	FK	
	Name, Email...	VARCHAR		
Roles	RoleID	INT	PK	e.g., Manager, Chef, Waiter
	RoleName	VARCHAR		

---

2. Relationships, Cardinality, and Participation

This section defines the business rules.

- *Cardinality:* How many records relate to how many? (1:1, 1:N, M:N)
- *Participation:* Is the relationship Mandatory (must have) or Optional (can have)?

Customers & Operations

1. *Customers ⇌ Orders*
   - *Cardinality:* One-to-Many (1:N)
   - *Participation:*
     - Customer Side: *Optional*. (A registered customer might not have ordered anything yet).
     - Order Side: *Mandatory*. (An order cannot exist without a customer).
2. *Customers ⇌ Reservations*
   - *Cardinality:* One-to-Many (1:N)
   - *Participation:*
     - Customer Side: *Optional*.
     - Reservation Side: *Mandatory*.
3. *Reservations ⇌ Tables*
   - *Cardinality:* Many-to-One (N:1) (A table can have many bookings over time).
   - *Participation:*

- Reservation Side: *Mandatory*. (Must book a specific table/table type).
- Table Side: *Optional*. (A table might currently have no future reservations).

Orders & Menu

4. *Orders ⇌ OrderItems*
   - *Cardinality:* One-to-Many (1:N)
   - *Participation:*
     - Order Side: *Mandatory*. (An order must contain at least one item).
     - Item Side: *Mandatory*. (An order item must belong to an order).
5. *MenuItems ⇌ OrderItems*
   - *Cardinality:* One-to-Many (1:N)
   - *Participation:*
     - MenuItem Side: *Optional*. (A newly added dish might not have been ordered yet).
     - OrderItem Side: *Mandatory*. (You cannot order "nothing"; it must be a valid menu item).

The Inventory Bridge

6. *MenuItems ⇌ RecipeIngredients*
   - *Cardinality:* One-to-Many (1:N)
   - *Participation:*
     - MenuItem Side: *Optional* (maybe a bottle of water has no "recipe").
     - Recipe Side: *Mandatory* (Every recipe entry must link to a menu item).
7. *InventoryItems ⇌ RecipeIngredients*
   - *Cardinality:* One-to-Many (1:N)
   - *Participation:*
     - Inventory Side: *Optional*. (We might stock napkins, which aren't in a "food recipe").
     - Recipe Side: *Mandatory*.

Supply Chain

8. *Suppliers ⇌ SupplyOrders*
   - *Cardinality:* One-to-Many (1:N)
   - *Participation:*
     - Supplier Side: *Optional*. (We added a supplier but haven't bought from them yet).
     - SupplyOrder Side: *Mandatory*.
9. *SupplyOrders ⇌ SupplyOrderItems*
   - *Cardinality:* One-to-Many (1:N)

- *Participation:* Both *Mandatory*.

Staff

10. *Roles ⇌ Staff*
    - *Cardinality:* One-to-Many (1:N)
    - *Participation:*
        - Role Side: *Optional* (A role like "Intern" exists but no one is hired for it yet).
        - Staff Side: *Mandatory* (Every employee must have a role).
11. *Staff ⇌ Orders*
    - *Cardinality:* One-to-Many (1:N)
    - *Participation:*
        - Staff Side: *Optional* (A new hire hasn't taken orders yet).
        - Order Side: *Mandatory* (Every order is processed by someone).

---

3. Summary of Normalization

Your schema is in *3rd Normal Form (3NF)*:

1. *1NF:* All attributes are atomic (no lists of items in a single cell).
2. *2NF:* No partial dependencies (bridge tables like OrderItems and RecipeIngredients solve this).
3. *3NF:* No transitive dependencies (e.g., The Orders table doesn't store the StaffEmail; it only stores StaffID. To get the email, you join the tables).

Would you like the SQL CREATE TABLE code for this final schema now?