Fields and relationship

To design the database schema for a pizza delivery app, you need to identify the key entities (tables) and the relationships between them. Here's a simplified example with the essential fields and relationships:

1. **User Table:**
    - Fields:
        - UserID (Primary Key)
        - Name
        - Email
        - Password (hashed)
        - Phone
        - Address
2. **Pizza Table:**
    - Fields:
        - PizzaID (Primary Key)
        - Name
        - Description
        - Price
        - Ingredients
3. **Order Table:**
    - Fields:
        - OrderID (Primary Key)
        - UserID (Foreign Key referencing User.UserID)
        - DeliveryAddress
        - OrderStatus (e.g., Pending, In Progress, Delivered)
        - TotalPrice
        - Timestamp
4. **OrderItem Table:**
    - Fields:
        - OrderItemID (Primary Key)

- OrderID (Foreign Key referencing Order.OrderID)
- PizzaID (Foreign Key referencing Pizza.PizzaID)
- Quantity

5. **Cart Table:**
   - Fields:
     - CartID (Primary Key)
     - UserID (Foreign Key referencing User.UserID)
6. **CartItem Table:**
   - Fields:
     - CartItemID (Primary Key)
     - CartID (Foreign Key referencing Cart.CartID)
     - PizzaID (Foreign Key referencing Pizza.PizzaID)
     - Quantity
7. **Payment Table:**
   - Fields:
     - PaymentID (Primary Key)
     - UserID (Foreign Key referencing User.UserID)
     - OrderID (Foreign Key referencing Order.OrderID)
     - Amount
     - PaymentStatus
     - PaymentMethod
8. **DeliveryDriver Table:**
   - Fields:
     - DriverID (Primary Key)
     - Name
     - VehicleInfo
     - CurrentLocation
9. **Notification Table:**
   - Fields:
     - NotificationID (Primary Key)
     - UserID (Foreign Key referencing User.UserID)
     - Message
     - Timestamp
10. **Restaurant Table:**
    - Fields:

- RestaurantID (Primary Key)
- Name
- Location
- Menu (Reference to Pizza.PizzaID)

Relationships:

- Each user can have multiple orders (one-to-many relationship between User and Order).
- Each order can have multiple items (one-to-many relationship between Order and OrderItem).
- Each user can have a cart, and each cart can contain multiple items (one-to-many relationship between User and Cart, and one-to-many relationship between Cart and CartItem).
- Each payment is associated with a user and an order (many-to-one relationship between Payment and User/Order).
- Each pizza in the menu can be part of multiple orders (many-to-many relationship between Pizza and Order through OrderItem).
- Each order can be assigned to a delivery driver (many-to-one relationship between Order and DeliveryDriver).
- Each user can receive multiple notifications (one-to-many relationship between User and Notification).

This is a simplified representation, and depending on the specific requirements of your pizza delivery app, you might need to adjust the schema. Additionally, consider using proper indexing, constraints, and normalization to ensure data integrity and efficient querying.