2024

Azure Cloud Integration and Database Design Purdue Global IT473 Prof. Savage

Prepared by: Moriah, Ian, Dalton, Chris

12/15/2024

Entity Relationship Model (ERM) Documentation PrimePathProvisions



**Entity Relationship Model Documentation for GotNoChill.com**

The Entity Relationship Model (ERM) for GotNoChill.com represents a comprehensive data structure designed to support a modern food delivery e-commerce platform targeting the 18-28 year old demographic. This model encompasses all critical aspects of the business operation, from user management to delivery tracking, ensuring seamless integration with our Azure cloud infrastructure.

**Core Entity Structure**

**User Management Domain**

At the center of our data model is the **USER** entity, which stores essential customer information, including authentication credentials and contact details. This entity forms relationships with multiple other entities, particularly **ADDRESS** and **ORDER**, allowing for flexible delivery options and order tracking. Separating **ADDRESS** into its own entity reflects our commitment to supporting multiple delivery locations per user, a crucial feature for our target demographic's mobile lifestyle.

**Product and Inventory Management**

The **PRODUCT** entity serves as the central repository for our food offerings, with detailed attributes including:

* Product specifications
* Pricing information
* Dietary information
* Stock management
* Expiration tracking

This entity connects directly to the **INVENTORY** entity, which maintains real-time stock levels and facilitates efficient warehouse management. The relationship between these entities enables accurate stock tracking while maintaining data normalization.

**Order Processing System**

The **ORDER** and **ORDER\_ITEM** entities form the backbone of our transaction processing system. The **ORDER** entity captures high-level transaction details, while **ORDER\_ITEM** maintains line-item specifics. This separation allows for:

* Flexible order composition
* Detailed order history
* Accurate pricing records
* Efficient order tracking

**Payment and Delivery Tracking**

The **PAYMENT** entity maintains secure transaction records while remaining separate from sensitive customer payment details, which will be handled by our payment processor. The **DELIVERY** entity tracks order fulfillment, including:

* Estimated delivery times
* Actual delivery confirmation
* Driver assignments
* Real-time status updates

**Relationships and Constraints**

Key relationships in our model include:

1. **One-to-Many**: Users to Orders (one user can place multiple orders)
2. **One-to-Many**: Orders to Order Items (one order can contain multiple items)
3. **One-to-One**: Orders to Payments (each order has exactly one payment record)
4. **One-to-One**: Orders to Deliveries (each order has one delivery record)
5. **Many-to-One**: Delivery to Driver (a single driver can handle multiple deliveries)

**ERM Diagram for PrimePath\_Provisions**

Link to LucidChart : <https://lucid.app/lucidchart/ecf392a6-b974-4bc6-ad98-ff86655d7c34/edit?viewport_loc=-874%2C-29%2C4020%2C1981%2C0_0&invitationId=inv_ed5f135d-4276-4354-b597-7ac2f15f7984>

A computer screen shot of a diagram

Description automatically generated

**ER Diagram SQL Code**

CREATE TABLE `User` (

`User\_ID` Int(PK),

`Email ` Varchar,

`Password\_hash` Varchar,

`First\_name` Varchar,

`Last\_name` Varchar,

`Phone` Varchar,

`Created\_at` Datetime,

`Is\_active` Boolean

);

CREATE TABLE `Order ` (

`Order\_id` Integer(PK),

`User\_id` INT(FK),

`Adress\_id` Int(FK),

`Status` Varchar,

`Total\_amount` Decimal(10,2),

`Payment\_status` Varchar,

`Delivery\_time` Datetime,

`Special\_in.` Text,

FOREIGN KEY (`Order\_id`) REFERENCES `User`(`Password\_hash`)

);

CREATE TABLE `Order\_Item` (

`Order\_itemID` Integer(PK),

`Order\_id` Int(FK),

`Product\_id` Int(FK),

`Quantity` Int,

`Unit\_price` decimal(10,2),

`Subtotal` Decimak(10,2)

);

CREATE TABLE `Address` (

`Address\_id` Int(PK),

`User\_id` Int(FK),

`Street` Varchar,

`City` Varchar,

`State` Varchar,

`Zip\_code` Varhcar,

`Delivery\_in.` Text,

`Is\_default` Boolean,

FOREIGN KEY (`Address\_id`) REFERENCES `User`(`Email `)

);

CREATE TABLE `Product` (

`Product\_id` Integer,

`Name` Varchar,

`Description` Text,

`Price` Decimal(10,2),

`Category` Varchar,

`Is\_available` Boolean,

`Stock\_quantity` Int,

`Dietary\_info` Text,

`Expiry\_date` Datetime

);

CREATE TABLE `Payment` (

`Payment\_id` Int(PK),

`Order\_id` Int(FK),

`Payment\_method` Varchar,

`Amount` Decimal(10,2),

`Transcation\_id` Varchar,

`Payment\_date` Datetime,

`Status` Varhcar

);

CREATE TABLE `Delivery` (

`Driver\_id` Int(PK),

`Full\_name` Varchar,

`Contact\_info` Varchar,

`Vehicle\_details` Varchar,

`Employment\_status` Varchar

);

CREATE TABLE `Inventory ` (

`Inventory\_id` INT(PK),

`Product\_id` Int(FK),

`Quantity` Int,

`Last\_update` Datetime,

`Storage\_location` Varhcar,

`Batch\_number` Varchar

);

CREATE TABLE `Driver ` (

`Driver\_id` Type,

`Full\_name ` Type,

`Contact\_info` Type,

`Vehicle\_details` Type

);

**Key Features of this ERM:**

**1. User Management**

* User profiles and authentication
* Multiple delivery addresses
* Order history tracking

**2. Product Management**

* Detailed product information
* Inventory tracking
* Category organization

**3. Order Processing**

* Order items and quantities
* Payment processing
* Delivery tracking

**4. Inventory Control**

* Stock management
* Batch tracking
* Storage location

**5. Delivery Management**

* Delivery scheduling
* Driver assignment
* Status tracking

**Example Relationships**

* **User → Address:** One-to-Many
* **User → Order:** One-to-Many
* **Order → Order\_Item:** One-to-Many
* **Order → Payment:** One-to-One
* **Order → Delivery:** One-to-One
* **Delivery → Driver:** Many-to-One
* **Order\_Item → Product:** Many-to-One
* **Product → Inventory:** One-to-One

The Entity Relationship Model (ERM) for GotNoChill.com provides a robust and scalable data structure designed to support a modern food delivery platform tailored to the needs of the 18-28-year-old demographic. Strategic separation of important business entities including **USER, ORDER, PRODUCT,** and **DELIVERY** guarantees effective management of user profiles, order processing, inventory control, and delivery tracking.   
The design of the model gives scalability, adaptability, and normalization the highest priority, thus allowing optimal connection with Azure cloud architecture. Key relationships **One-to- Many** links for user orders and **One-to-One** links for payments and deliveries show a clear, orderly method of data flow and transaction management.  
Apart from supporting real-time corporate operations, this ERM provides a strong basis for upcoming developments including sophisticated analytics, automation, and better user experience. The GotNoChill.com platform is fit to provide effective, reliable, and premium food delivery services to its target audience with its comprehensive entity relationships and logical constraints.