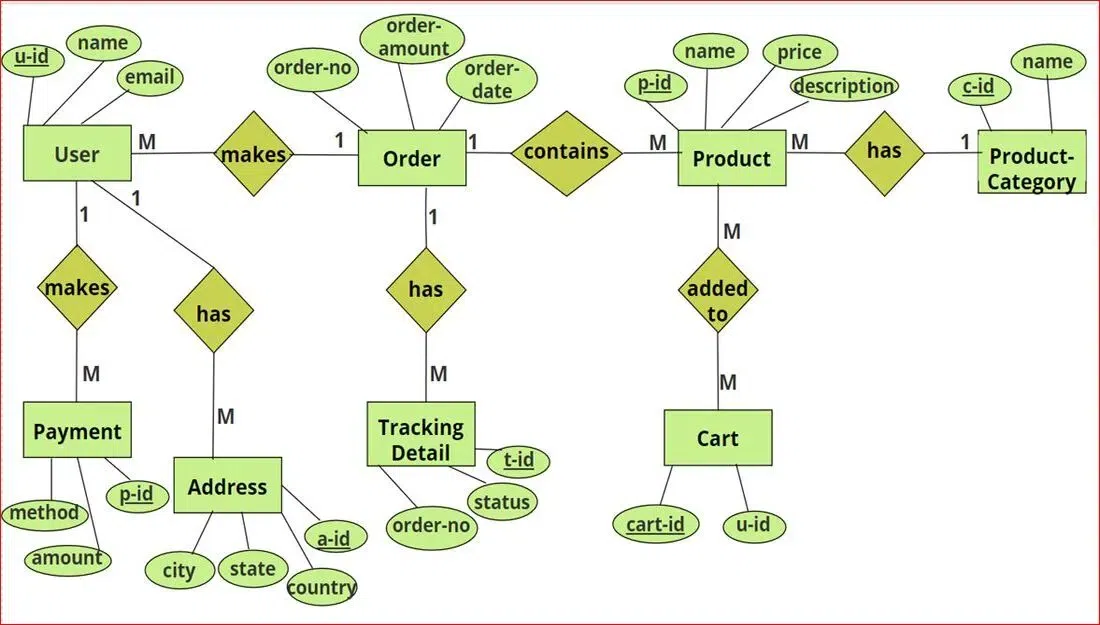
**ER-Diagram** :**Grandma’sBliss - Sweets and Pickles**

Group :28

DS-Beta

**ER-Diagram :**



**Collections :**

1. **users –** Stores user details
2. **orders –** Stores order details
3. **products –** Stores product details
4. **product\_categories –** Stores product category details
5. **carts –** Stores cart details
6. **payments –** Stores payment details
7. **addresses –** Stores user address details
8. **tracking\_details –** Stores order tracking details

**Entities & Attributes:**

1. **User**
   * Attributes: u-id, name, email
   * Relationships:
     + Makes **Orders** (1:M)
     + Has **Payment** details (1:M)
     + Has **Address** (1:M)
     + Associated with **Cart** (1:1)
2. **Order**
   * Attributes: order-no, order-amount, order-date
   * Relationships:
     + Made by a **User** (M:1)
     + Contains **Products** (1:M)
     + Has **Tracking Details** (1:M)
     + Linked to an **Address** (M:1)
3. **Product**
   * Attributes: p-id, name, price, description
   * Relationships:
     + Contained in an **Order** (M:1)
     + Added to a **Cart** (M:M)
     + Belongs to a **Product-Category** (M:1)
4. **Product-Category**
   * Attributes: c-id, name
   * Relationships:
     + Contains multiple **Products** (1:M)
5. **Cart**
   * Attributes: cart-id, u-id
   * Relationships:
     + Contains multiple **Products** (M:M)
     + Belongs to a **User** (1:1)
6. **Payment**
   * Attributes: p-id, method, amount
   * Relationships:
     + Belongs to a **User** (M:1)
7. **Address**
   * Attributes: a-id, city, state, country
   * Relationships:
     + Linked to multiple **Orders** (1:M)
     + Belongs to a **User** (M:1)
8. **Tracking Detail**
   * Attributes: t-id, order-no, status
   * Relationships:
     + Belongs to an **Order** (M:1)

**Relationships:**

**User ↔ Orders** (One-to-Many)

* A user can place multiple orders, but each order belongs to only one user.
* This is managed by storing the user\_id reference in the orders collection.

**Order ↔ Products** (Many-to-Many)

* An order can contain multiple products, and a product can be part of multiple orders.
* This is implemented using an array of product references inside the orders collection.

klpj**Product ↔ Product Category** (Many-to-One)

* Each product belongs to a single category, but a category can have multiple products.
* The products collection contains a reference to the product\_categories collection.

**User ↔ Cart** (One-to-One)

* Each user has exactly one cart.
* The carts collection contains a user\_id reference to associate it with a user.

**Cart ↔ Products** (Many-to-Many)

* A cart can have multiple products, and a product can be in multiple carts.
* This is stored as an embedded array of products inside the carts collection.

**User ↔ Addresses** (One-to-Many)

* A user can have multiple addresses, but each address belongs to only one user.
* The addresses collection includes a user\_id reference.

**Order ↔ Address** (Many-to-One)

* Each order is shipped to a specific address, but multiple orders can use the same address.
* The orders collection contains an address\_id reference.

**User ↔ Payments** (One-to-Many)

* A user can have multiple payment transactions, but each payment is linked to only one user.
* The payments collection contains a user\_id reference.

**Order ↔ Tracking Details** (One-to-Many)

* Each order can have multiple tracking updates over time.
* The tracking\_details collection contains an order\_id reference to link it with an order.