

NoSQL Data Models - AI Integrated Grocery List Application

Table of Contents

1. [Collections Overview](#)
2. [Schema Definitions](#)
3. [Validation Rules](#)
4. [Example Documents](#)
5. [Indexes](#)
6. [Common Queries](#)

Collections Overview

The application uses the following collections:

- users: Stores user profiles and preferences
- shoppingLists: Contains shopping lists and items
- products: Product catalog information
- mealPlans: User meal planning data
- retailers: Store information and layouts
- aiRecommendations: AI-generated suggestions and patterns

Relationships

```
users —|> shoppingLists
      |> mealPlans
      |> aiRecommendations

products <—> retailers
        |> shoppingLists
```

Schema Definitions

users

```
{
  _id: ObjectId,
  email: String,           // Required, unique
  passwordHash: String,   // Required
  firstName: String,       // Required
  lastName: String,        // Required
  preferences: {
    budgetLimit: Number,
    notificationsEnabled: Boolean,
    preferredStore: ObjectId,
    dietaryRestrictions: [{
      type: String,
      description: String
    }]
  },
  createdAt: Date,
  lastLogin: Date
}
```

shoppingLists

```
{
  _id: ObjectId,
  userId: ObjectId,        // Reference to users
  name: String,            // Required
  items: [{
    productId: ObjectId,   // Reference to products
    productName: String,
    quantity: Number,
    price: Number,
    category: String,
    isChecked: Boolean,
    notes: String
  }],
  isActive: Boolean,
  createdAt: Date,
  modifiedAt: Date
}
```

products

```
{
  _id: ObjectId,
  name: String,           // Required
  description: String,
  category: {
    name: String,
    department: String
  },
  retailers: [{
    retailerId: ObjectId,
    price: Number,
    isAvailable: Boolean,
    aisleLocation: String,
    barcode: String
  }],
  nutritionalInfo: {
    calories: Number,
    protein: Number,
    allergens: [String]
  },
  lastUpdated: Date
}
```

mealPlans

```
{
  _id: ObjectId,
  userId: ObjectId,       // Reference to users
  name: String,
  startDate: Date,
  endDate: Date,
  meals: [{
    day: Number,
    mealType: String,
    products: [{
      productId: ObjectId,
      productName: String,
      quantity: Number
    }]
  }],
  createdAt: Date
}
```

retailers

```
{
  _id: ObjectId,
  name: String,           // Required
  location: {
    address: String,
    city: String,
    postalCode: String
  },
  storeLayout: {
    aisles: [{
      aisleNumber: String,
      section: String,
      categories: [String]
    }]
  },
  contactInfo: {
    phone: String,
    email: String
  }
}
```

aiRecommendations

```
{
  _id: ObjectId,
  userId: ObjectId,       // Reference to users
  recommendationType: String,
  recommendations: [{
    productId: ObjectId,
    productName: String,
    confidence: Number,
    reason: String,
    lastPurchased: Date
  }],
  userPatterns: {
    frequentItems: [String],
    preferredCategories: [String],
    averageSpending: Number
  },
  createdAt: Date
}
```

Validation Rules

users Collection

```
{
  validator: {
    $jsonSchema: {
      bsonType: "object",
      required: ["email", "passwordHash", "firstName", "lastName"],
      properties: {
        email: {
          bsonType: "string",
          pattern: "^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$"
        },
        passwordHash: {
          bsonType: "string",
          minLength: 60
        },
        preferences: {
          bsonType: "object",
          properties: {
            budgetLimit: {
              bsonType: "number",
              minimum: 0
            }
          }
        }
      }
    }
  }
}
```

shoppingLists Collection

```
{
  validator: {
    $jsonSchema: {
      bsonType: "object",
      required: ["userId", "name", "items"],
      properties: {
        items: {
          bsonType: "array",
          items: {
            bsonType: "object",
            required: ["productId", "quantity"]
          }
        }
      }
    }
  }
}
```