**Cosmos DB Setup and Queries for Foodie Database**

**1. Cosmos DB Account and Database Setup Instructions**

Step 1: Go to the Azure Portal.

Step 2: Create a new Cosmos DB account with SQL API.

Step 3: Enable Free Tier.

Step 4: Create a new database with the ID foodie.

**2. Container Creation with Partition and Hierarchical Keys**

|  |  |  |
| --- | --- | --- |
| **Container** | **Partition Key** | **Hierarchical Partition Key** |
| CATEGORIES | /CategoryId | None |
| PRODUCTS | /CategoryId | /ProductId |
| CART | /UserId | /ProductId |
| USERS | /UserId | None |
| ORDERS | /UserId | /OrderId |
| PAYMENTS | /UserId | /PaymentId |
| OrderDetails | /OrderId | /ProductId |
| CONTACT | /ContactId | None |

For each container, in the "Partition key" field, you can set up hierarchical partition keys as follows:

* **For PRODUCTS**: /CategoryId/ProductId
* **For CART**: /UserId/ProductId
* **For ORDERS**: /UserId/OrderId
* **For PAYMENTS**: /UserId/PaymentId
* **For OrderDetails**: /OrderId/ProductId

**1. CATEGORIES Table**

* Container Name: Categories
* Partition Key: /CategoryId
* Schema: Json

{ "CategoryId": "int",

"Name": "string",

"ImageUrl": "string",

"IsActive": "boolean",

"CreatedDate": "datetime"

}

**2. PRODUCTS Table**

* Container Name: Products
* Partition Key: /ProductId
* Schema: json

{

"ProductId": "int",

"Name": "string",

"Description": "string",

"Price": "decimal",

"Quantity": "int",

"ImageUrl": "string",

"CategoryId": "int", // Foreign Key reference to `Categories`

"IsActive": "boolean",

"CreatedDate": "datetime"

}

**3. CART Table**

* Container Name: Cart
* Partition Key: /CartId
* Schema: json

{

"CartId": "int",

"ProductId": "int", // Foreign Key reference to `Products`

"Quantity": "int",

"UserId": "int" // Foreign Key reference to `Users`

}

**4. USERS Table**

* Container Name: Users
* Partition Key: /UserId
* Schema: json

{

"UserId": "int",

"Name": "string",

"Username": "string", // Unique

"Email": "string",

"Address": "string",

"PostCode": "string",

"Mobile": "string"

}

**5. ORDERS Table**

* Container Name: Orders
* Partition Key: /OrderDetailsId
* Schema: json

{

"OrderDetailsId": "int",

"OrderNo": "string", // Unique

"ProductId": "int", // Foreign Key reference to `Products`

"Quantity": "int",

"UserId": "int", // Foreign Key reference to `Users`

"Status": "string",

"PaymentId": "int", // Foreign Key reference to `Payment`

"OrderDate": "datetime"

}

**6. CONTACT Table**

* Container Name: Contact
* Partition Key: /ContactId
* Schema: json

{

"ContactId": "int",

"Name": "string",

"Email": "string",

"Subject": "string",

"Message": "string",

"CreatedDate": "datetime"

}

**7. PAYMENT Table**

* Container Name: Payment
* Partition Key: /PaymentI
* Schema:

{

"PaymentId": "int",

"Name": "string",

"CardNo": "string",

"ExpiryDate": "string",

"CvvNo": "int",

"Address": "string",

"PaymentMode": "string"

}