# Food Order App - NoSQL Database Design, Insert and Retrieval

**Tasks 1**

1. Screenshot for users collection

|  |
| --- |
|  |

1. Screenshot for userSessions collection

|  |
| --- |
|  |

1. Screenshot for categories collection

|  |
| --- |
|  |

1. Screenshot for cuisines collection

|  |
| --- |
|  |

1. Screenshot for restaurants collection

|  |
| --- |
|  |

1. Screenshot for fooditems collection

|  |
| --- |
|  |

1. Screenshot for menus collection

|  |
| --- |
|  |

1. Screenshot for menuItems collection

|  |
| --- |
|  |

1. Screenshot for carts collections

|  |
| --- |
|  |

1. Screenshot for cartItems collection

|  |
| --- |
|  |

1. Screenshot for shippingDetails collection

|  |
| --- |
|  |

1. Screenshot for orders collection

|  |
| --- |
|  |

1. Screenshot for orderItems collection

|  |
| --- |
|  |

**Task 2**

1. All distinct category names

|  |
| --- |
| db.categories.distinct("name"); |

1. All vegetarian fooditems

|  |
| --- |
| db.foodItems.find({isVeg:true}) |

1. Retrieve the name and address of each of the restaurants.

|  |
| --- |
| db.restaurants.find({}, { \_id:0, name: 1, address: 1 }) |

1. All orders with orderTotalPrice within the range of $75 to $150.

|  |
| --- |
| db.orders.find({orderTotalPrice : {$gte : 75, $lte : 150}}) |

1. All orders which were created after 10th May 2023.

|  |
| --- |
| db.orders.find({createdTs : {$gte : ISODate('2023-05-10')}}) |

1. All fooditems belonging to Italian cuisine.

|  |
| --- |
| db.fooditems.find({ cuisineId: db.cuisines.findOne({ name: "Italian" }).id }) |

1. Retrieve all menu items available in the restaurant named “Gourmet Junction”.

|  |
| --- |
| db.menuitems.find({ menuId: db.menus.findOne({ restaurantId: db.restaurants.findOne({ name: "Gourmet Junction" }).id }) }); ({ name: "Gourmet Junction" }).id }) }); |

1. Retrieve details of the order with the lowest orderTotalPrice.

|  |
| --- |
| db.orders.findOne({}, {}, { sort: { orderTotalPrice: 1 } }) |

1. Get the count of menu items for fooditem “Veg Pizza”.

|  |
| --- |
| db.menuitems.countDocuments({ fooditemId: db.fooditems.findOne({name:"Veg Pizza"}) }) |

1. Retrieve id of the cart having maximum no. of fooditems in it.

|  |
| --- |
| db.cartItems.aggregate([ { $sortByCount: "$cartId" }, { $limit: 1 }, { $project: { \_id: 0, cartId: "$\_id" } }]) |