AIT Food Delivery System

Content

Importance of this Project

- The idea of Food delivery system is to deliver food made by the students to the people within the AIT campus, while saving their time from going to restaurants and carrying food. In today's world, students and teachers are highly busy with their work, and sometimes get very less time to go out for food or even miss out on their lunch or dinner time due to work or assignment pressure..

- The Food delivery system saves their time by placing order for them and delivering food to their dorms, homes or workplaces, which helps them to focus on their work as well as get food in the right time.

- The major stakeholders include the students/staffs who want to sell their food, and the customers ordering food.

- This system would also help people to understand the different types of cuisines that are available around them, different students from various nations can cook and sell their native foods.

What makes the project appealing

- The project is highly appealing, since it focuses on the food delivery system at only university campus level.

- It saves time for the stakeholders by providing food for them in due time from their homes or workplace, while continuing their work. The platform will be very beneficial for both the faculty members as well as the students of the university.

- Finally, the platform will create good source of income to the students or other staff members who are willing to sell their foods, as well as create a good source of revenue for the university.

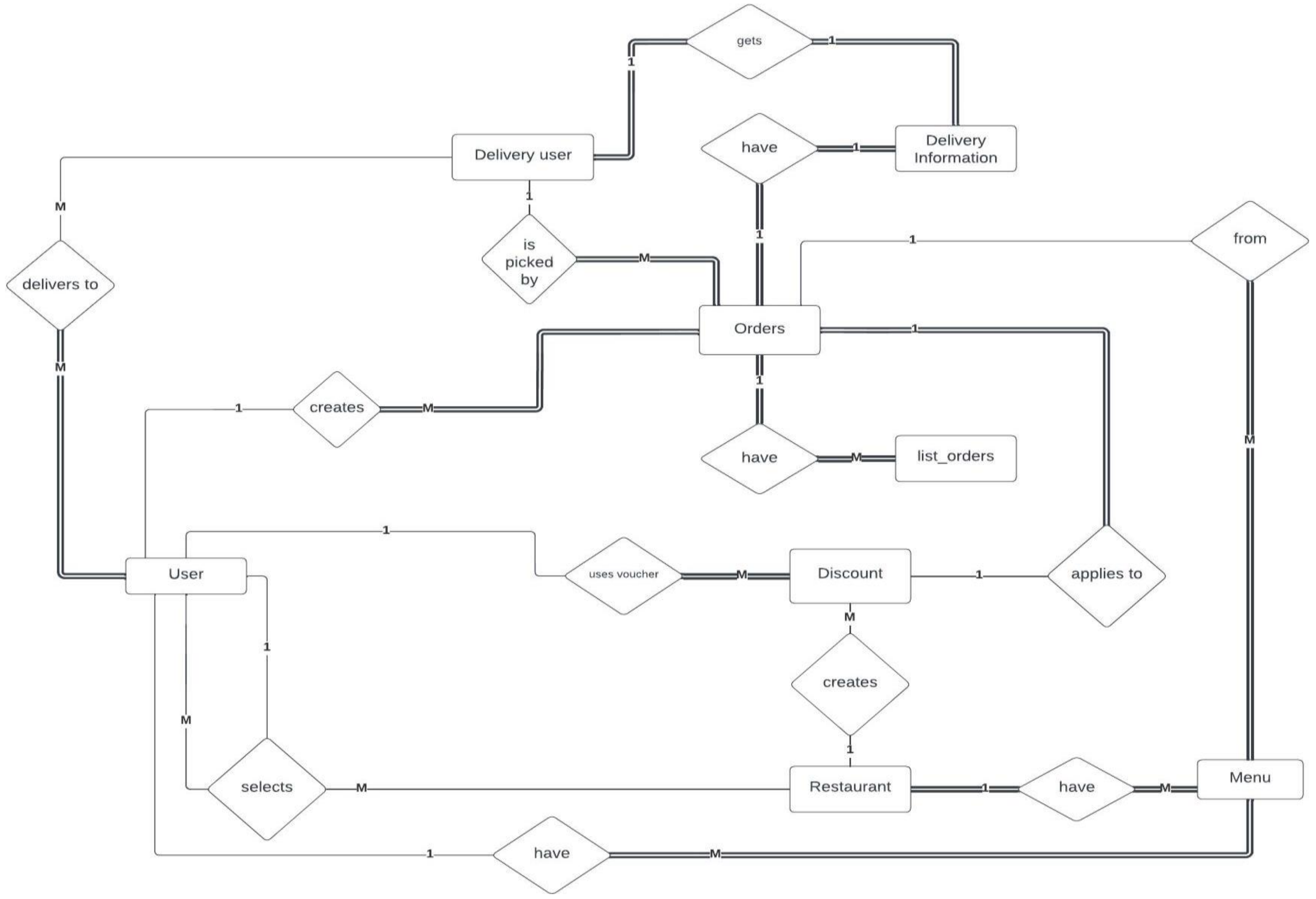
Business Rules and Constraints

- The food delivery system is for the AIT campus community.
- Different types of foods can be uploaded to the app by the sellers.
- Customers can select different food from different sellers to order multiple types of food through different orders.
- Customers can pay using COD, QR scan and in built wallet.
- Customers can login using their mobile phone and otp for verification. Providing an address is not needed for registering
- Sellers can select their order hours, based on their availability.

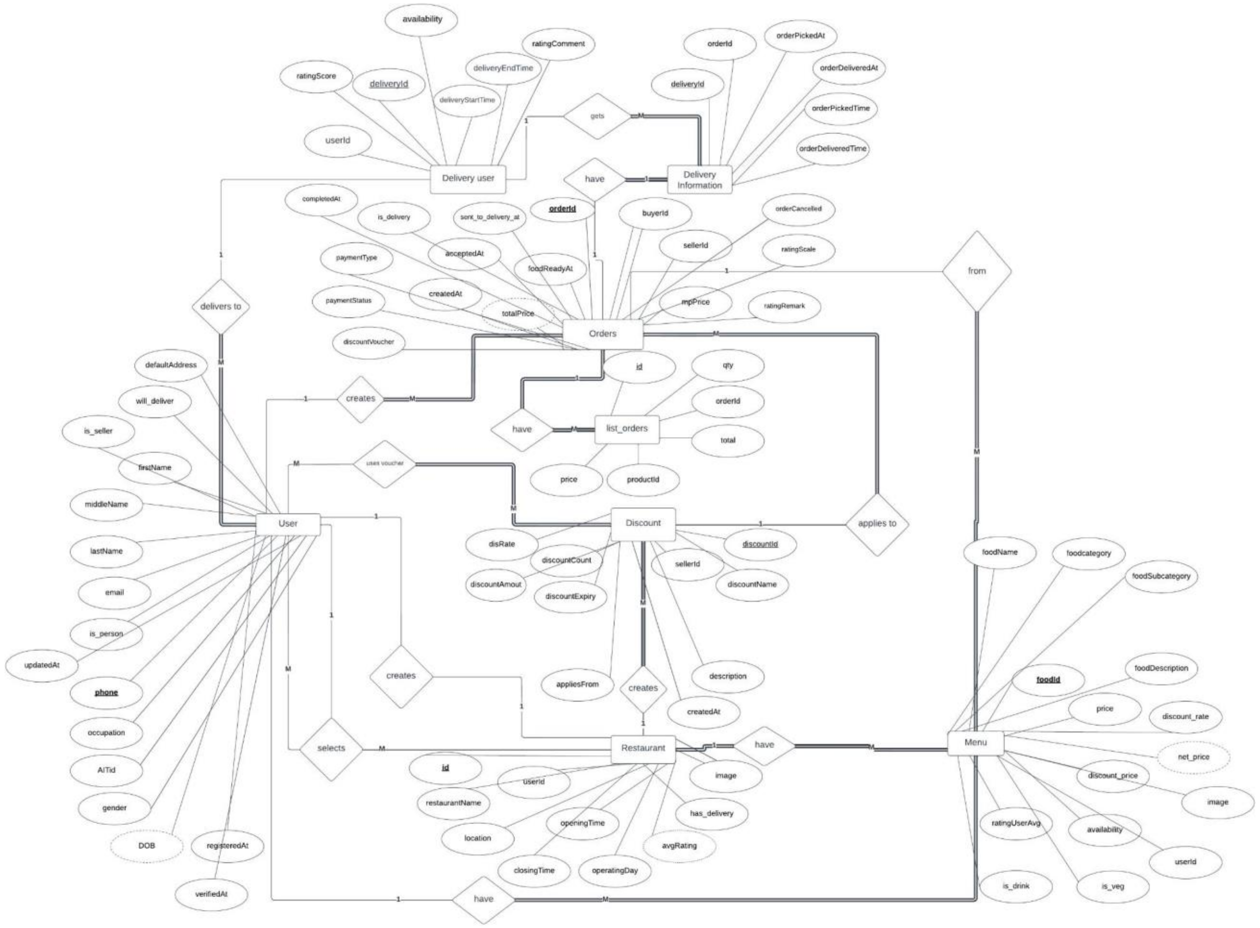
Business Rules and Constraints

- Customers can also avail gift vouchers/ discount options.
- Customers can provide ratings to sellers regarding food. This will help to rate the sellers.
- Sellers cannot delete any food from the menu if they have a pending order.
- For multiple orders, invoices will be one invoice as per seller.
- The system will keep track of the delivery status.

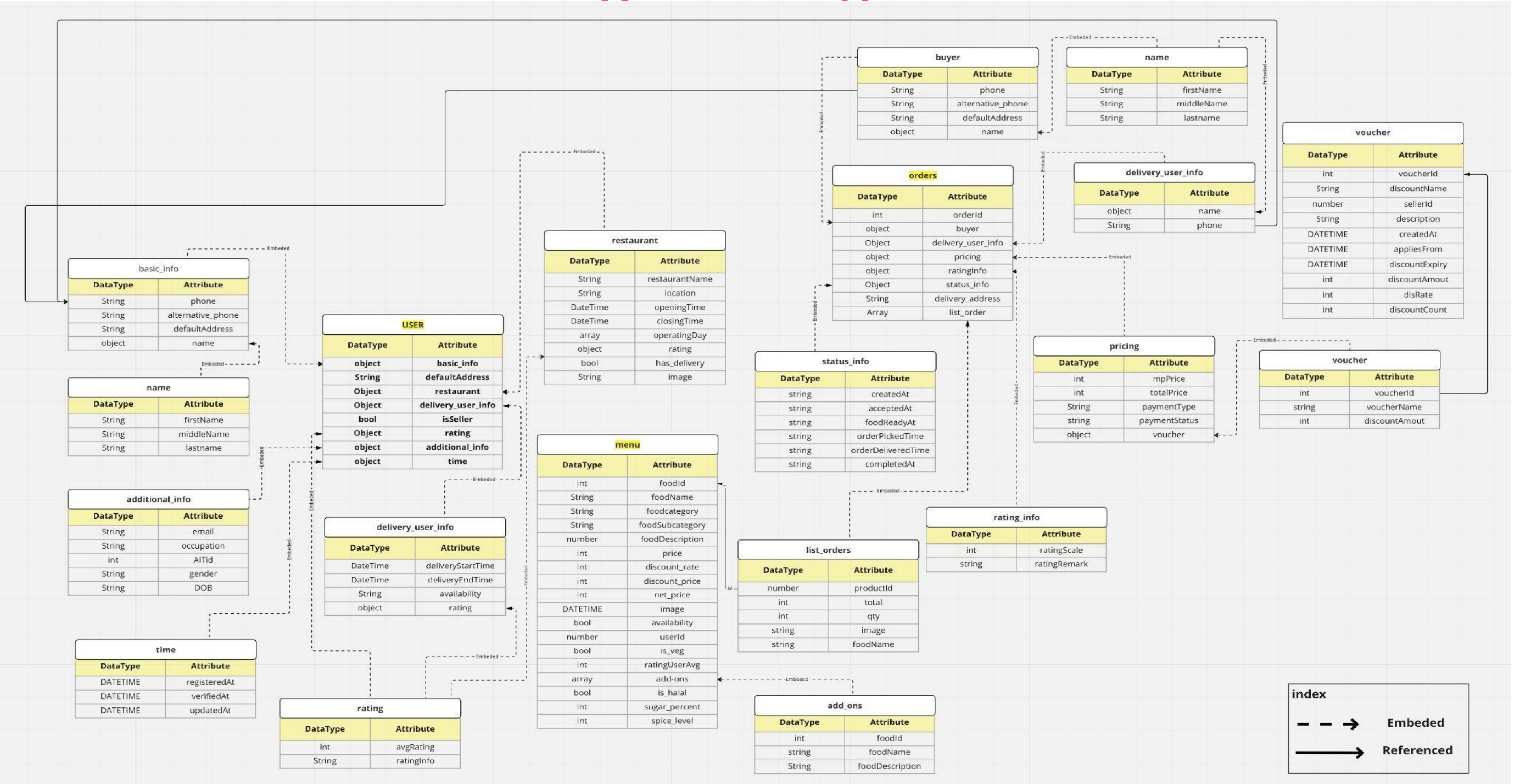
Conceptual Diagram - Overview



Conceptual Diagram



Logical Diagram



buyer

DataType	Attribute
String	phone
String	alternative_phone
String	defaultAddress
object	name

name

DataType	Attribute
String	firstName
String	middleName
String	lastname

voucher

DataType	Attribute
int	voucherId
String	discountName
number	sellerId
String	description
DATETIME	createdAt
DATETIME	appliesFrom
DATETIME	discountExpiry
int	discountAmout
int	disRate
int	discountCount

orders

DataType	Attribute
int	orderId
object	buyer
Object	delivery_user_info
object	pricing
object	ratingInfo
Object	status_info
String	delivery_address
Array	list_order

delivery_user_info

DataType	Attribute
object	name
String	phone

restaurant

DataType	Attribute
String	restaurantName
String	location
DateTime	openingTime
DateTime	closingTime
array	operatingDay
object	rating
bool	has_delivery
String	image

basic_info

DataType	Attribute
String	phone
String	alternative_phone
String	defaultAddress
object	name

USER

DataType	Attribute
object	basic_info
String	defaultAddress
Object	restaurant
Object	delivery_user_info
bool	isSeller
Object	rating
object	additional_info
object	time

name

DataType	Attribute
String	firstName
String	middleName
String	lastname

status_info

DataType	Attribute
string	createdAt
string	acceptedAt
string	foodReadyAt
string	orderPickedTime
string	orderDeliveredTime
string	completedAt

pricing

DataType	Attribute
int	mpPrice
int	totalPrice
String	paymentType
string	paymentStatus
object	voucher

voucher

DataType	Attribute
int	voucherId
string	voucherName
int	discountAmout

menu

DataType	Attribute
int	foodId
String	foodName
String	foodcategory
String	foodSubcategory
number	foodDescription
int	price
int	discount_rate
int	discount_price
int	net_price
DATETIME	image
bool	availability
number	userId
bool	is_veg
int	ratingUserAvg
array	add-ons
bool	is_halal
int	sugar_percent
int	spice_level

additional_info

DataType	Attribute
String	email
String	occupation
int	AITid
String	gender
String	DOB

delivery_user_info

DataType	Attribute
DateTime	deliveryStartTime
DateTime	deliveryEndTime
String	availability
object	rating

rating_info

DataType	Attribute
int	ratingScale
string	ratingRemark

list_orders

DataType	Attribute
number	productId
int	total
int	qty
string	image
string	foodName

time

DataType	Attribute
DATETIME	registeredAt
DATETIME	verifiedAt
DATETIME	updatedAt

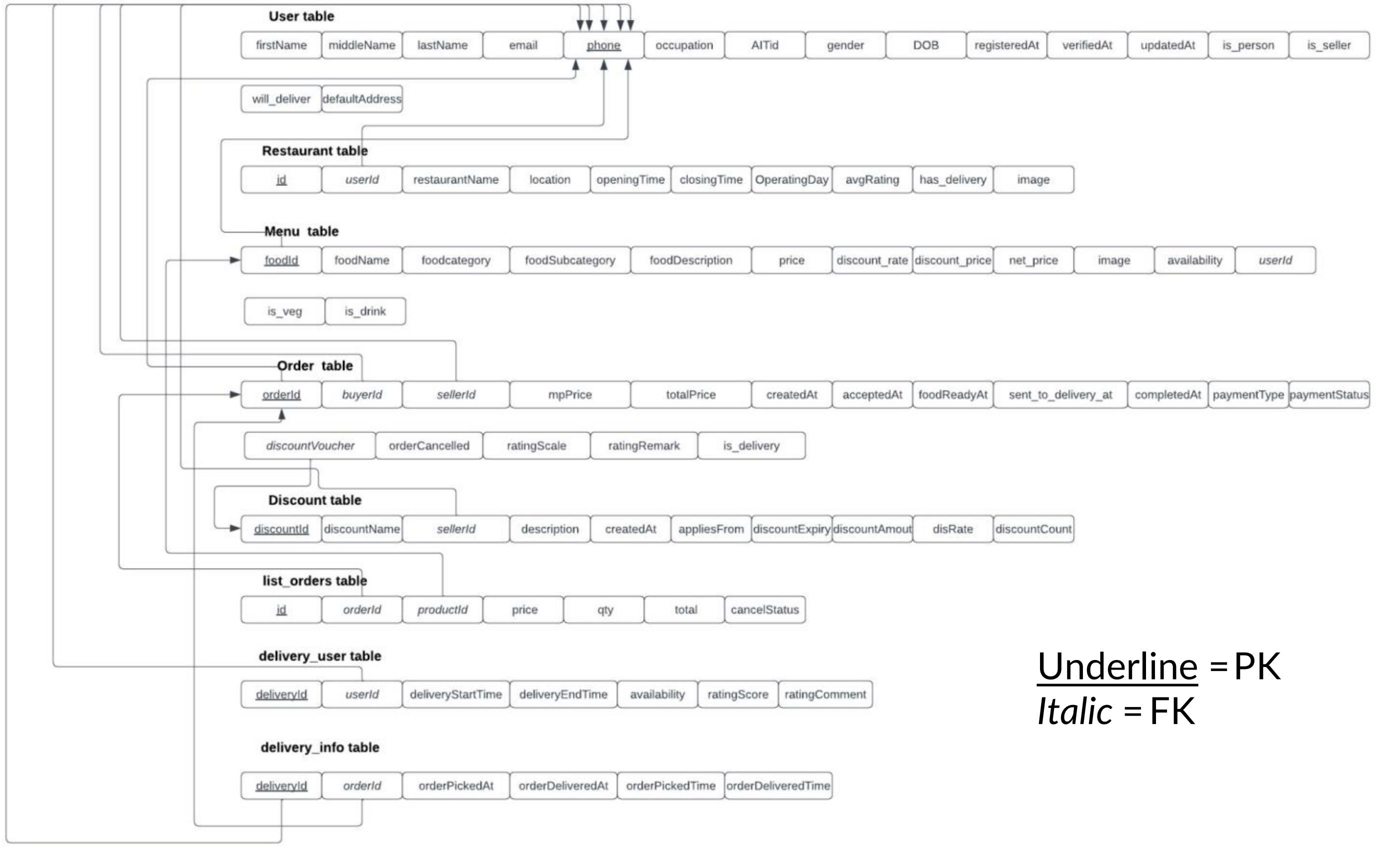
rating

DataType	Attribute
int	avgRating
String	ratingInfo

add_ons

DataType	Attribute
int	foodId
string	foodName
String	foodDescription

Logical Diagram



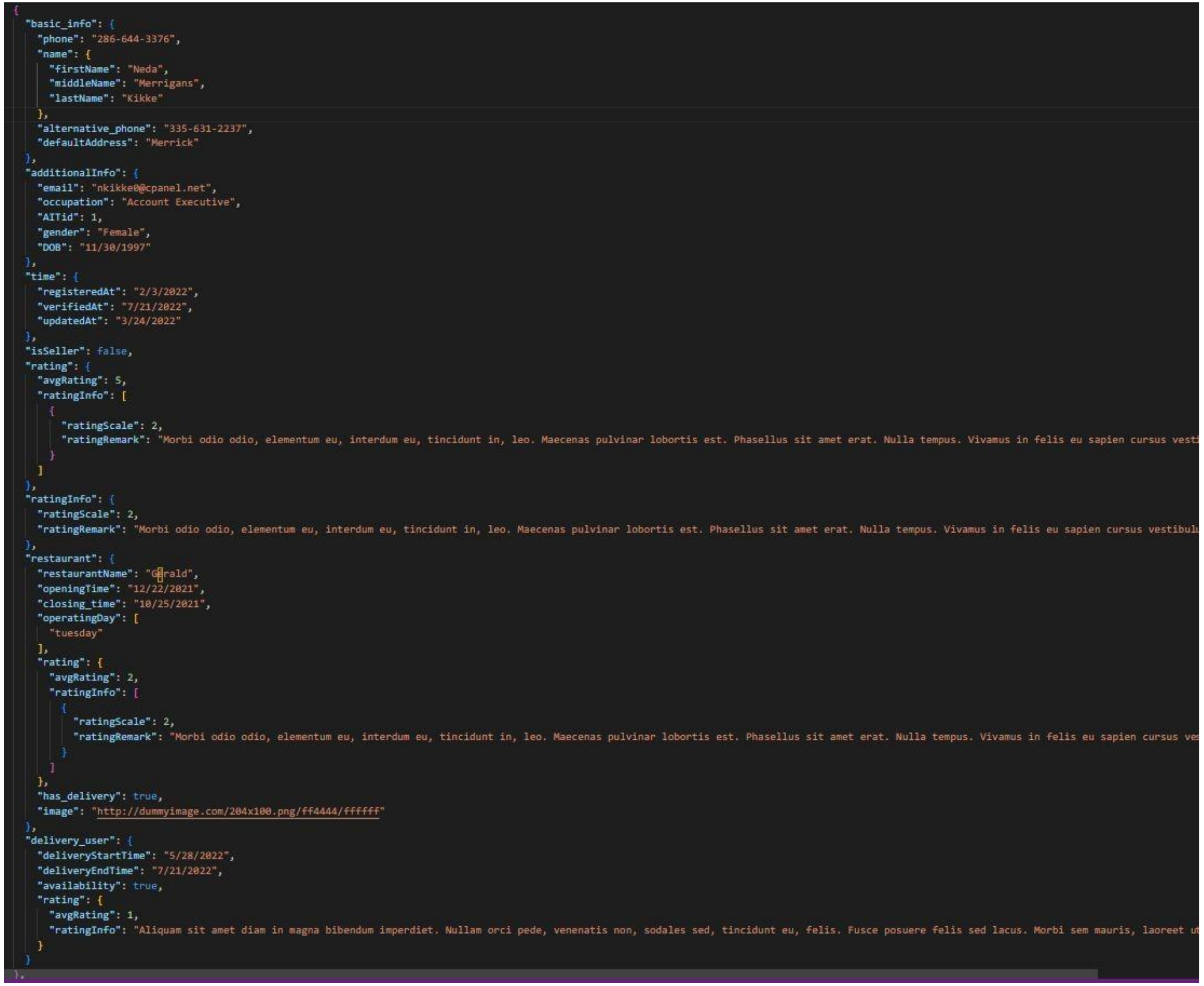
User table
firstName | middleName | lastName | email | phone | occupation | AITid | gender | DOB | registeredAt | verifiedAt | updatedAt | is_person | is_seller
will_deliver | defaultAddress
Restaurant table
id | *userId* | restaurantName | location | openingTime | closingTime | OperatingDay | avgRating | has_delivery | image
Menu table
foodId | foodName | foodcategory | foodSubcategory | foodDescription | price | discount_rate | discount_price | net_price | image | availability | *userId*
is_veg | is_drink
Order table
orderId | *buyerId* | *sellerId* | mpPrice | totalPrice | createdAt | acceptedAt | foodReadyAt | sent_to_delivery_at | completedAt | paymentType | paymentStatus
discountVoucher | orderCancelled | ratingScale | ratingRemark | is_delivery
Discount table
discountId | discountName | *sellerId* | description | createdAt | appliesFrom | discountExpiry | discountAmout | disRate | discountCount
list_orders table
id | *orderId* | *productId* | price | qty | total | cancelStatus
delivery_user table
deliveryId | *userId* | deliveryStartTime | deliveryEndTime | availability | ratingScore | ratingComment
delivery_info table
deliveryId | *orderId* | orderPickedAt | orderDeliveredAt | orderPickedTime | orderDeliveredTime

Underline = PK
Italic = FK

Important Queries and Operations of NoSQL

Identifying important insert, update and delete operations of NoSQL in MongoDB

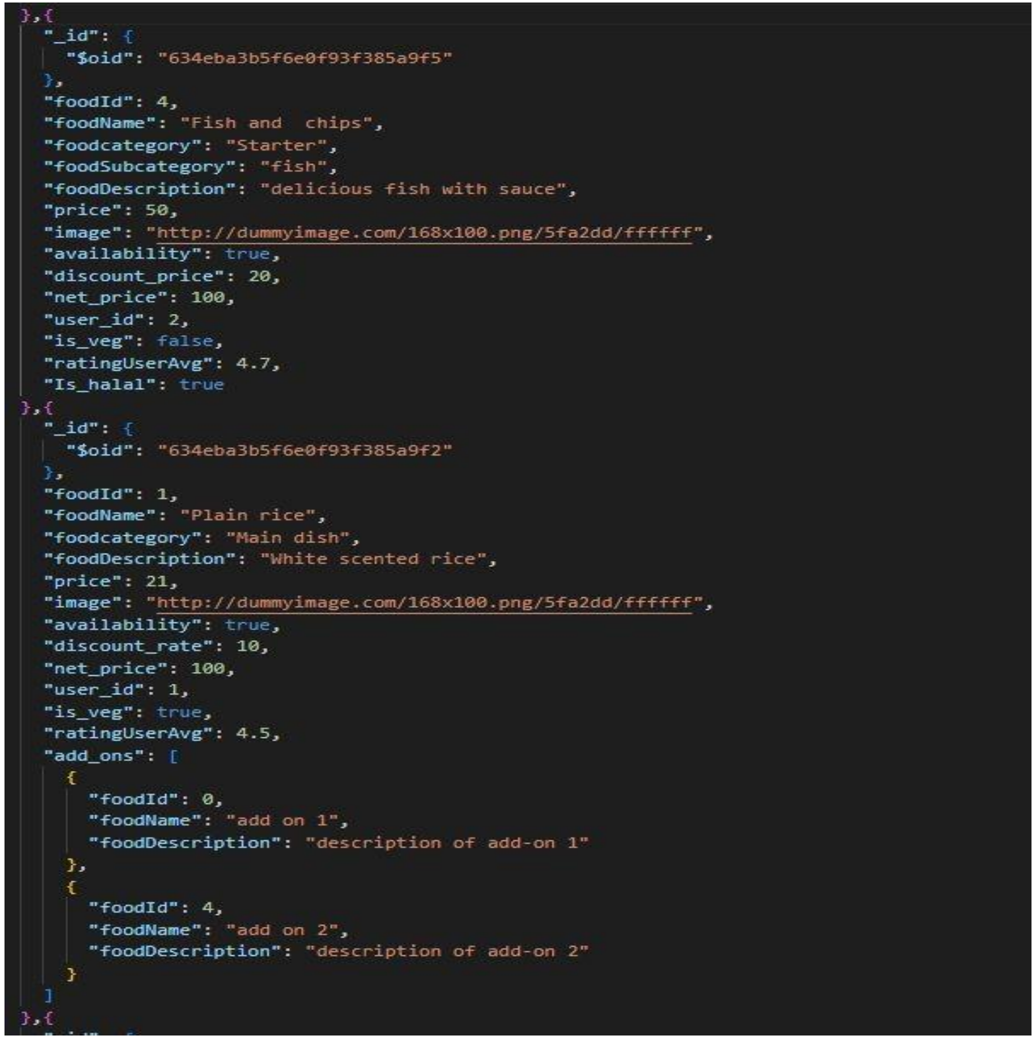
- Inserting a new user in user collection
- Inserting new food item in menu collection
- Checking availability status of food in menu collection
- Checking discount in voucher table
- Checking order status of each orders
- Checking reviews from customers by menu of restaurant
- Inserting new discounts voucher given by sellers



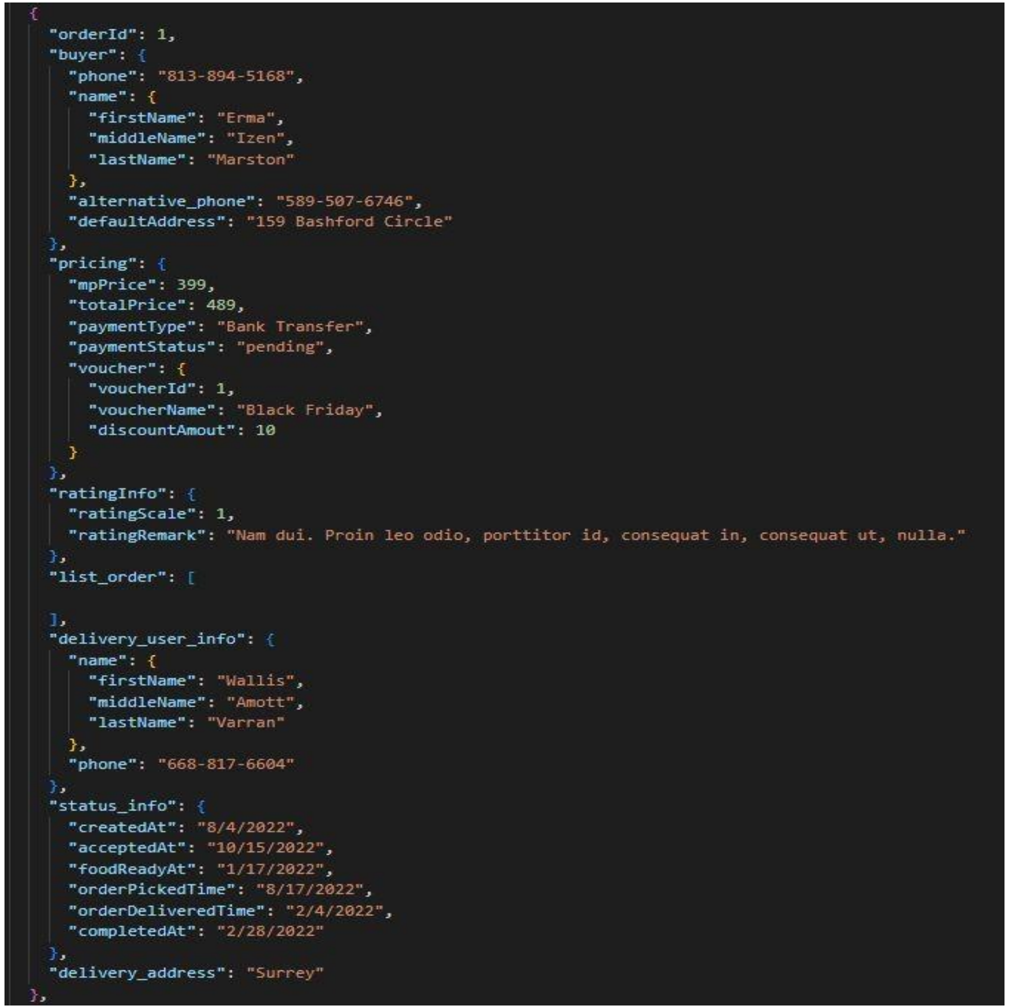
```
{
  "basic_info": {
    "phone": "286-644-3376",
    "name": {
      "firstName": "Neda",
      "middleName": "Merrigans",
      "lastName": "Kikke"
    },
    "alternative_phone": "335-631-2237",
    "defaultAddress": "Merrick"
  },
  "additionalInfo": {
    "email": "nkikke0@cpanel.net",
    "occupation": "Account Executive",
    "AITid": 1,
    "gender": "Female",
    "DOB": "11/30/1997"
  },
  "time": {
    "registeredAt": "2/3/2022",
    "verifiedAt": "7/21/2022",
    "updatedAt": "3/24/2022"
  },
  "isSeller": false,
  "rating": {
    "avgRating": 5,
    "ratingInfo": [
      {
        "ratingScale": 2,
        "ratingRemark": "Morbi odio odio, elementum eu, interdum eu, tincidunt in, leo. Maecenas pulvinar lobortis est. Phasellus sit amet erat. Nulla tempus. Vivamus in felis eu sapien cursus vesti
      }
    ]
  },
  "ratingInfo": {
    "ratingScale": 2,
    "ratingRemark": "Morbi odio odio, elementum eu, interdum eu, tincidunt in, leo. Maecenas pulvinar lobortis est. Phasellus sit amet erat. Nulla tempus. Vivamus in felis eu sapien cursus vestibul
  },
  "restaurant": {
    "restaurantName": "Gerald",
    "openingTime": "12/22/2021",
    "closing_time": "10/25/2021",
    "operatingDay": [
      "tuesday"
    ],
    "rating": {
      "avgRating": 2,
      "ratingInfo": [
        {
          "ratingScale": 2,
          "ratingRemark": "Morbi odio odio, elementum eu, interdum eu, tincidunt in, leo. Maecenas pulvinar lobortis est. Phasellus sit amet erat. Nulla tempus. Vivamus in felis eu sapien cursus ves
        }
      ]
    },
    "has_delivery": true,
    "image": "http://dummyimage.com/204x100.png/ff4444/ffffff"
  },
  "delivery_user": {
    "deliveryStartTime": "5/28/2022",
    "deliveryEndTime": "7/21/2022",
    "availability": true,
    "rating": {
      "avgRating": 1,
      "ratingInfo": "Aliquam sit amet diam in magna bibendum imperdiet. Nullam orci pede, venenatis non, sodales sed, tincidunt eu, felis. Fusce posuere felis sed lacus. Morbi sem mauris, laoreet ut
    }
  }
},
```

User Collection

Menu Collection



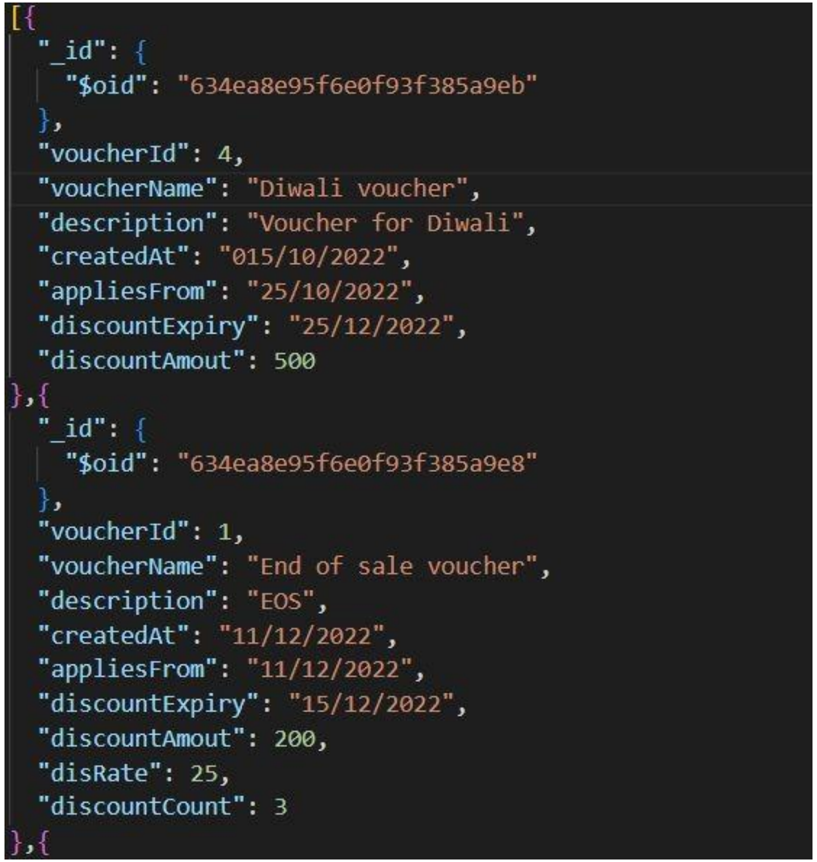
```
},{
  "_id": {
    "$oid": "634eba3b5f6e0f93f385a9f5"
  },
  "foodId": 4,
  "foodName": "Fish and  chips",
  "foodcategory": "Starter",
  "foodSubcategory": "fish",
  "foodDescription": "delicious fish with sauce",
  "price": 50,
  "image": "http://dummyimage.com/168x100.png/5fa2dd/ffffff",
  "availability": true,
  "discount_price": 20,
  "net_price": 100,
  "user_id": 2,
  "is_veg": false,
  "ratingUserAvg": 4.7,
  "Is_halal": true
},{
  "_id": {
    "$oid": "634eba3b5f6e0f93f385a9f2"
  },
  "foodId": 1,
  "foodName": "Plain rice",
  "foodcategory": "Main dish",
  "foodDescription": "White scented rice",
  "price": 21,
  "image": "http://dummyimage.com/168x100.png/5fa2dd/ffffff",
  "availability": true,
  "discount_rate": 10,
  "net_price": 100,
  "user_id": 1,
  "is_veg": true,
  "ratingUserAvg": 4.5,
  "add_ons": [
    {
      "foodId": 0,
      "foodName": "add on 1",
      "foodDescription": "description of add-on 1"
    },
    {
      "foodId": 4,
      "foodName": "add on 2",
      "foodDescription": "description of add-on 2"
    }
  ]
},{
```



```
{
  "orderId": 1,
  "buyer": {
    "phone": "813-894-5168",
    "name": {
      "firstName": "Erma",
      "middleName": "Izen",
      "lastName": "Marston"
    },
    "alternative_phone": "589-507-6746",
    "defaultAddress": "159 Bashford Circle"
  },
  "pricing": {
    "mpPrice": 399,
    "totalPrice": 489,
    "paymentType": "Bank Transfer",
    "paymentStatus": "pending",
    "voucher": {
      "voucherId": 1,
      "voucherName": "Black Friday",
      "discountAmout": 10
    }
  },
  "ratingInfo": {
    "ratingScale": 1,
    "ratingRemark": "Nam dui. Proin leo odio, porttitor id, consequat in, consequat ut, nulla."
  },
  "list_order": [

  ],
  "delivery_user_info": {
    "name": {
      "firstName": "Wallis",
      "middleName": "Amott",
      "lastName": "Varran"
    },
    "phone": "668-817-6604"
  },
  "status_info": {
    "createdAt": "8/4/2022",
    "acceptedAt": "10/15/2022",
    "foodReadyAt": "1/17/2022",
    "orderPickedTime": "8/17/2022",
    "orderDeliveredTime": "2/4/2022",
    "completedAt": "2/28/2022"
  },
  "delivery_address": "Surrey"
},
```

Order Collection

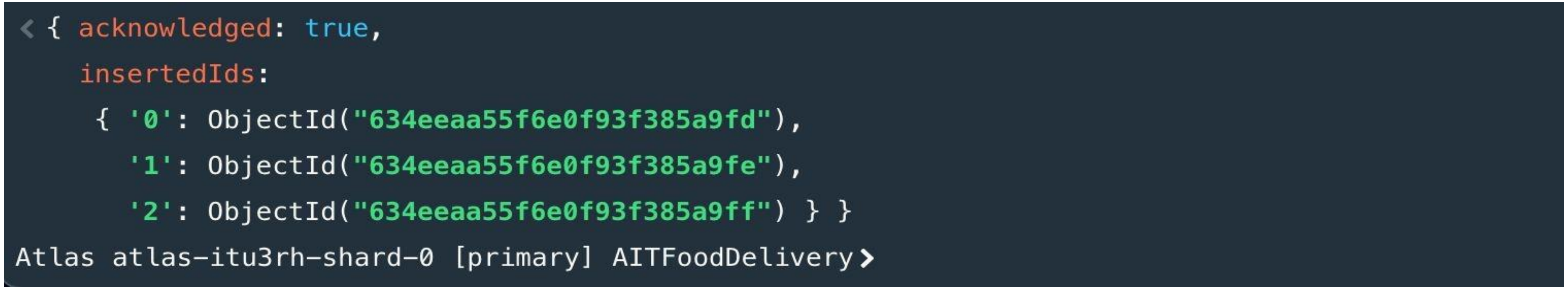


```
[{
  "_id": {
    "$oid": "634ea8e95f6e0f93f385a9eb"
  },
  "voucherId": 4,
  "voucherName": "Diwali voucher",
  "description": "Voucher for Diwali",
  "createdAt": "015/10/2022",
  "appliesFrom": "25/10/2022",
  "discountExpiry": "25/12/2022",
  "discountAmout": 500
},{
  "_id": {
    "$oid": "634ea8e95f6e0f93f385a9e8"
  },
  "voucherId": 1,
  "voucherName": "End of sale voucher",
  "description": "EOS",
  "createdAt": "11/12/2022",
  "appliesFrom": "11/12/2022",
  "discountExpiry": "15/12/2022",
  "discountAmout": 200,
  "disRate": 25,
  "discountCount": 3
},{
```

Voucher Collection

Important Queries and Operations of NoSQL

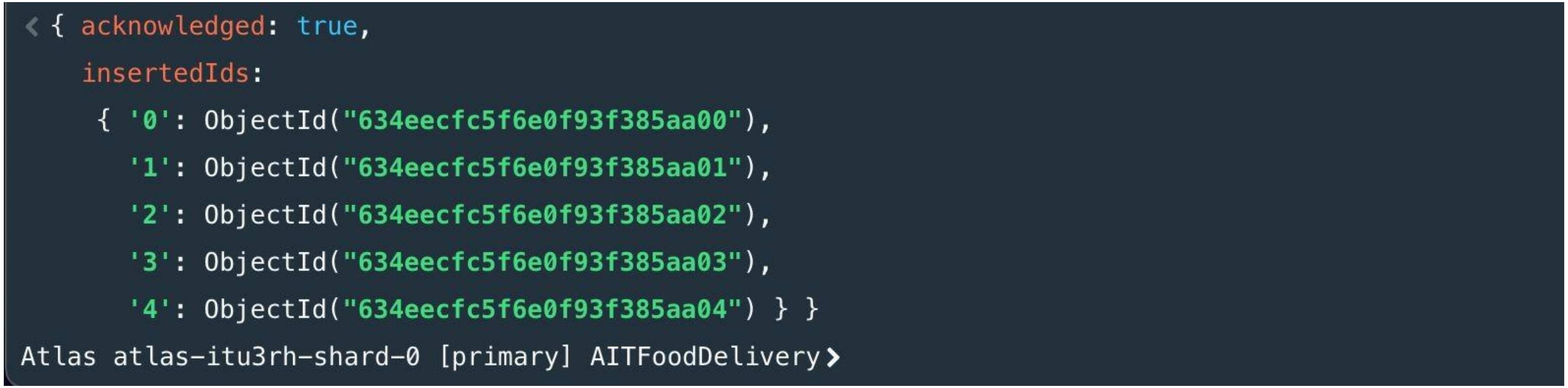
New users are registered in the user collection



```
< { acknowledged: true,
    insertedIds:
     { '0': ObjectId("634eeaa55f6e0f93f385a9fd"),
       '1': ObjectId("634eeaa55f6e0f93f385a9fe"),
       '2': ObjectId("634eeaa55f6e0f93f385a9ff") } }
Atlas atlas-itu3rh-shard-0 [primary] AITFoodDelivery>
```

Important Queries and Operations

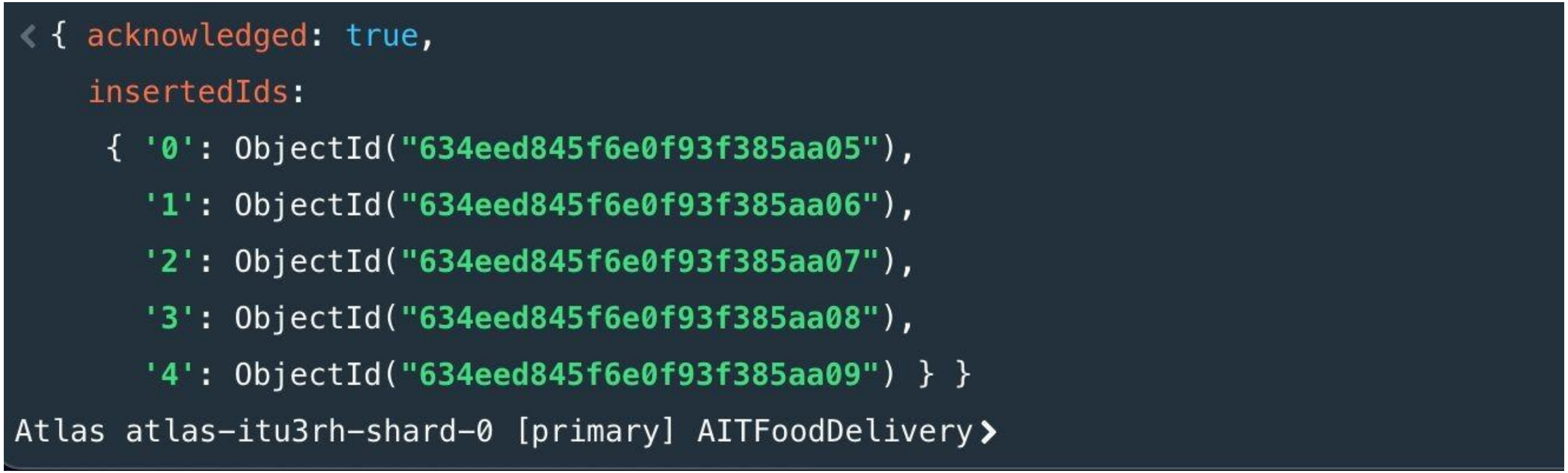
New vouchers are added into the voucher collection



```
< { acknowledged: true,
    insertedIds:
     { '0': ObjectId("634eecfc5f6e0f93f385aa00"),
       '1': ObjectId("634eecfc5f6e0f93f385aa01"),
       '2': ObjectId("634eecfc5f6e0f93f385aa02"),
       '3': ObjectId("634eecfc5f6e0f93f385aa03"),
       '4': ObjectId("634eecfc5f6e0f93f385aa04") } }
Atlas atlas-itu3rh-shard-0 [primary] AITFoodDelivery>
```

Important Queries and Operations of NoSQL

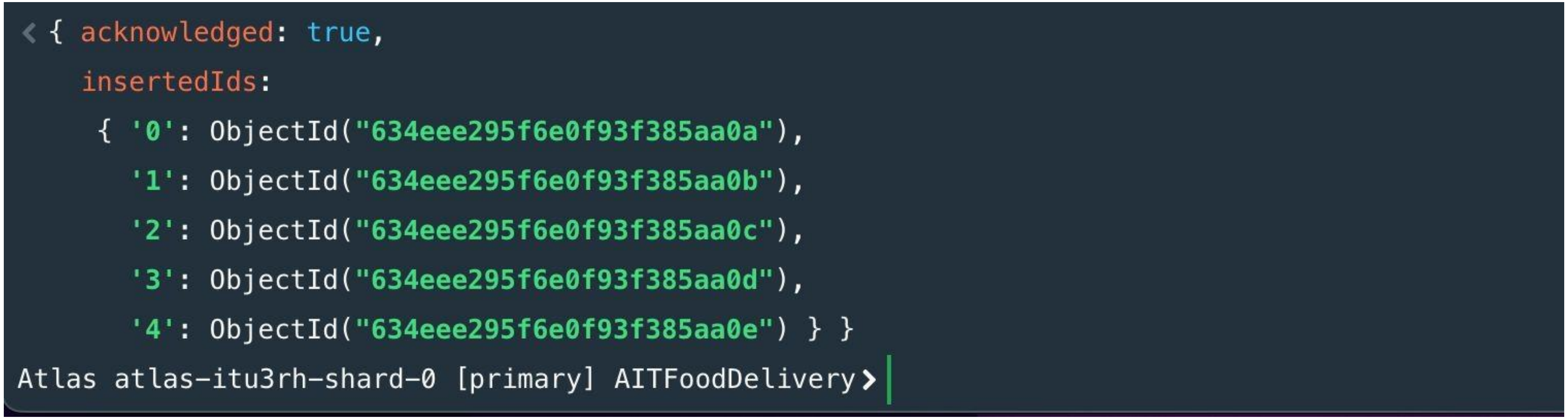
New orders are placed into the order collections



```
< { acknowledged: true,
    insertedIds:
     { '0': ObjectId("634eed845f6e0f93f385aa05"),
       '1': ObjectId("634eed845f6e0f93f385aa06"),
       '2': ObjectId("634eed845f6e0f93f385aa07"),
       '3': ObjectId("634eed845f6e0f93f385aa08"),
       '4': ObjectId("634eed845f6e0f93f385aa09") } }
Atlas atlas-itu3rh-shard-0 [primary] AITFoodDelivery>
```

Important Queries and Operations of NoSQL

New menus from different restaurants/sellers are added into the menu collection

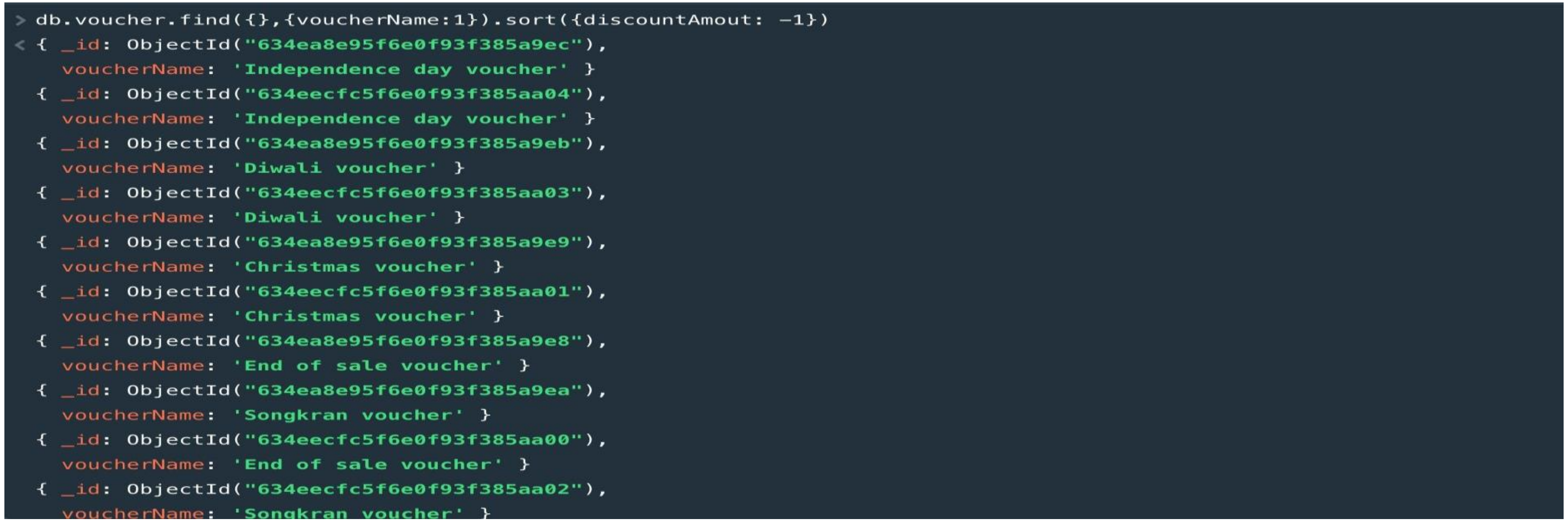


```
< { acknowledged: true,
    insertedIds:
     { '0': ObjectId("634eee295f6e0f93f385aa0a"),
       '1': ObjectId("634eee295f6e0f93f385aa0b"),
       '2': ObjectId("634eee295f6e0f93f385aa0c"),
       '3': ObjectId("634eee295f6e0f93f385aa0d"),
       '4': ObjectId("634eee295f6e0f93f385aa0e") } }
Atlas atlas-itu3rh-shard-0 [primary] AITFoodDelivery>
```

Business queries of NoSQL

Showing top festivals or occasions where discount has been given highest to lowest.

```
db.voucher.find({},{voucherName:1}).sort({discountAmout: -1})
```

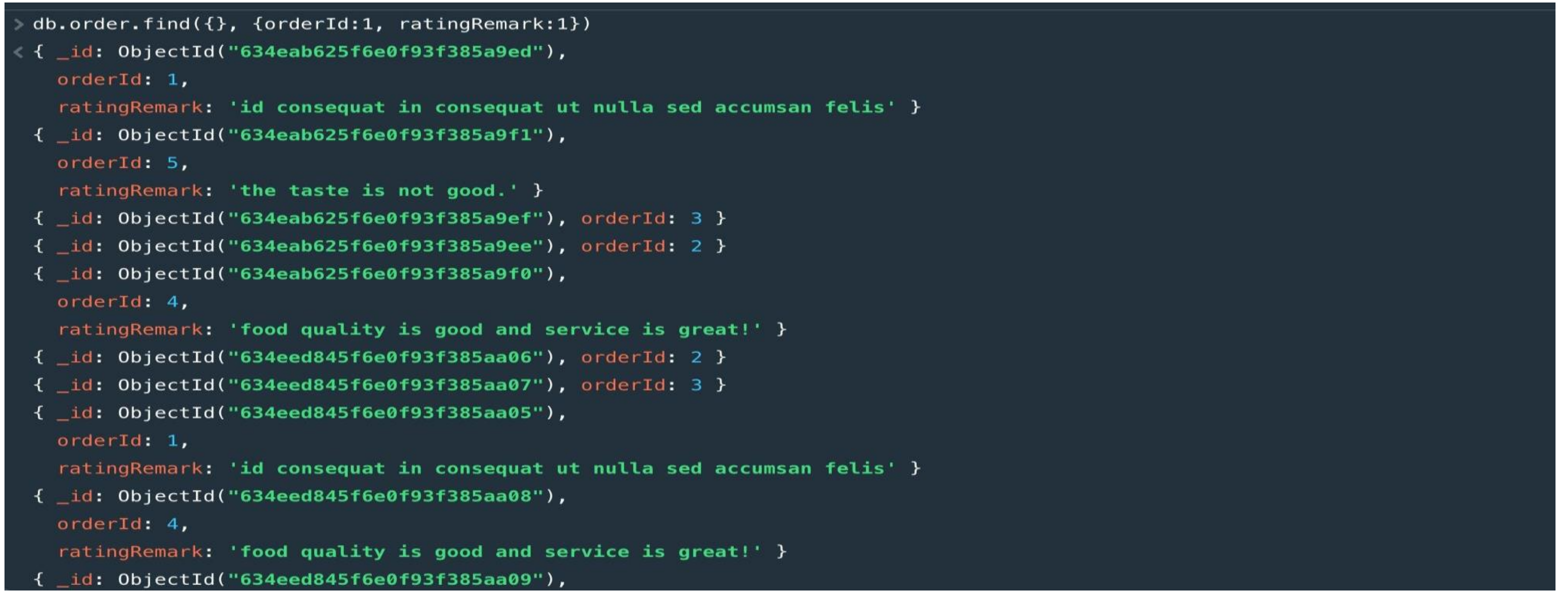


```
> db.voucher.find({},{voucherName:1}).sort({discountAmout: -1})
< { _id: ObjectId("634ea8e95f6e0f93f385a9ec"),
    voucherName: 'Independence day voucher' }
  { _id: ObjectId("634eecfc5f6e0f93f385aa04"),
    voucherName: 'Independence day voucher' }
  { _id: ObjectId("634ea8e95f6e0f93f385a9eb"),
    voucherName: 'Diwali voucher' }
  { _id: ObjectId("634eecfc5f6e0f93f385aa03"),
    voucherName: 'Diwali voucher' }
  { _id: ObjectId("634ea8e95f6e0f93f385a9e9"),
    voucherName: 'Christmas voucher' }
  { _id: ObjectId("634eecfc5f6e0f93f385aa01"),
    voucherName: 'Christmas voucher' }
  { _id: ObjectId("634ea8e95f6e0f93f385a9e8"),
    voucherName: 'End of sale voucher' }
  { _id: ObjectId("634ea8e95f6e0f93f385a9ea"),
    voucherName: 'Songkran voucher' }
  { _id: ObjectId("634eecfc5f6e0f93f385aa00"),
    voucherName: 'End of sale voucher' }
  { _id: ObjectId("634eecfc5f6e0f93f385aa02"),
    voucherName: 'Songkran voucher' }
```

Business queries of NoSQL

Showing the rating remarks of customers on their orders.

```
db.order.find({}, {orderId:1, ratingRemark:1})
```

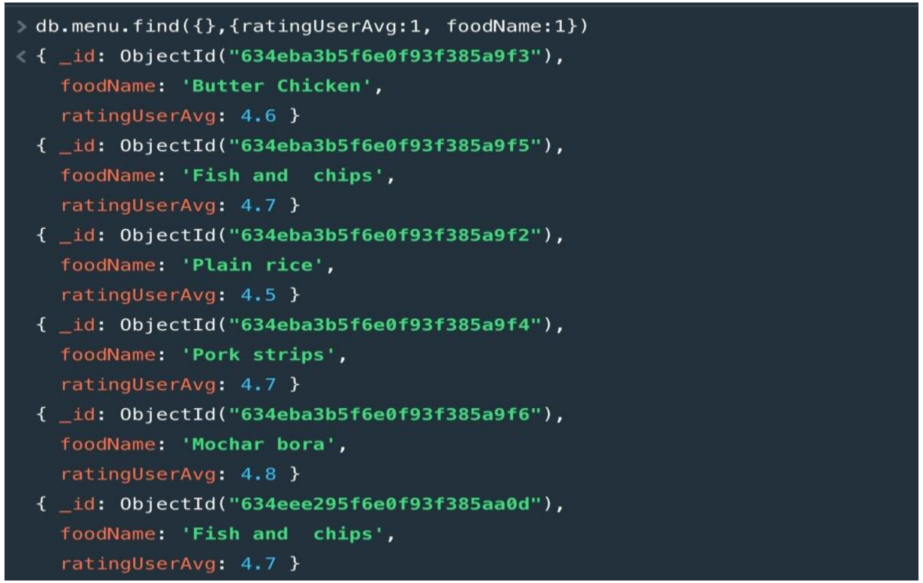


```
> db.order.find({}, {orderId:1, ratingRemark:1})
< { _id: ObjectId("634eab625f6e0f93f385a9ed"),
    orderId: 1,
    ratingRemark: 'id consequat in consequat ut nulla sed accumsan felis' }
  { _id: ObjectId("634eab625f6e0f93f385a9f1"),
    orderId: 5,
    ratingRemark: 'the taste is not good.' }
  { _id: ObjectId("634eab625f6e0f93f385a9ef"), orderId: 3 }
  { _id: ObjectId("634eab625f6e0f93f385a9ee"), orderId: 2 }
  { _id: ObjectId("634eab625f6e0f93f385a9f0"),
    orderId: 4,
    ratingRemark: 'food quality is good and service is great!' }
  { _id: ObjectId("634eed845f6e0f93f385aa06"), orderId: 2 }
  { _id: ObjectId("634eed845f6e0f93f385aa07"), orderId: 3 }
  { _id: ObjectId("634eed845f6e0f93f385aa05"),
    orderId: 1,
    ratingRemark: 'id consequat in consequat ut nulla sed accumsan felis' }
  { _id: ObjectId("634eed845f6e0f93f385aa08"),
    orderId: 4,
    ratingRemark: 'food quality is good and service is great!' }
  { _id: ObjectId("634eed845f6e0f93f385aa09"),
```

Business queries of NoSQL

Showing the rating given by users on the food.

```
db.menu.find({},{ratingUserAvg:1, foodName:1})
```

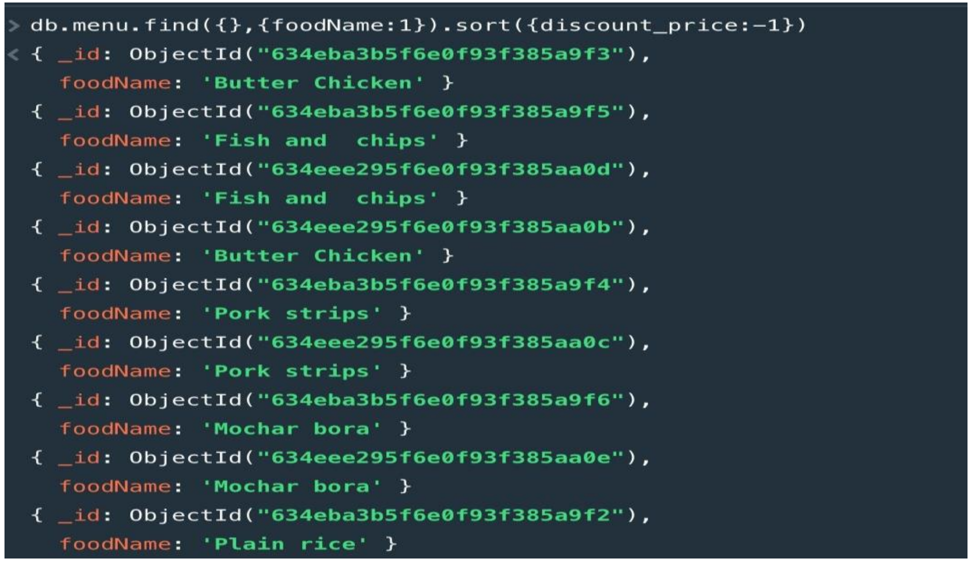


```
> db.menu.find({},{ratingUserAvg:1, foodName:1})
< { _id: ObjectId("634eba3b5f6e0f93f385a9f3"),
    foodName: 'Butter Chicken',
    ratingUserAvg: 4.6 }
  { _id: ObjectId("634eba3b5f6e0f93f385a9f5"),
    foodName: 'Fish and  chips',
    ratingUserAvg: 4.7 }
  { _id: ObjectId("634eba3b5f6e0f93f385a9f2"),
    foodName: 'Plain rice',
    ratingUserAvg: 4.5 }
  { _id: ObjectId("634eba3b5f6e0f93f385a9f4"),
    foodName: 'Pork strips',
    ratingUserAvg: 4.7 }
  { _id: ObjectId("634eba3b5f6e0f93f385a9f6"),
    foodName: 'Mochar bora',
    ratingUserAvg: 4.8 }
  { _id: ObjectId("634eee295f6e0f93f385aa0d"),
    foodName: 'Fish and  chips',
    ratingUserAvg: 4.7 }
```

Business queries of NoSQL

Finding most discount food from menu

```
db.menu.find({},{foodName:1}).sort({discount_price:-1})
```

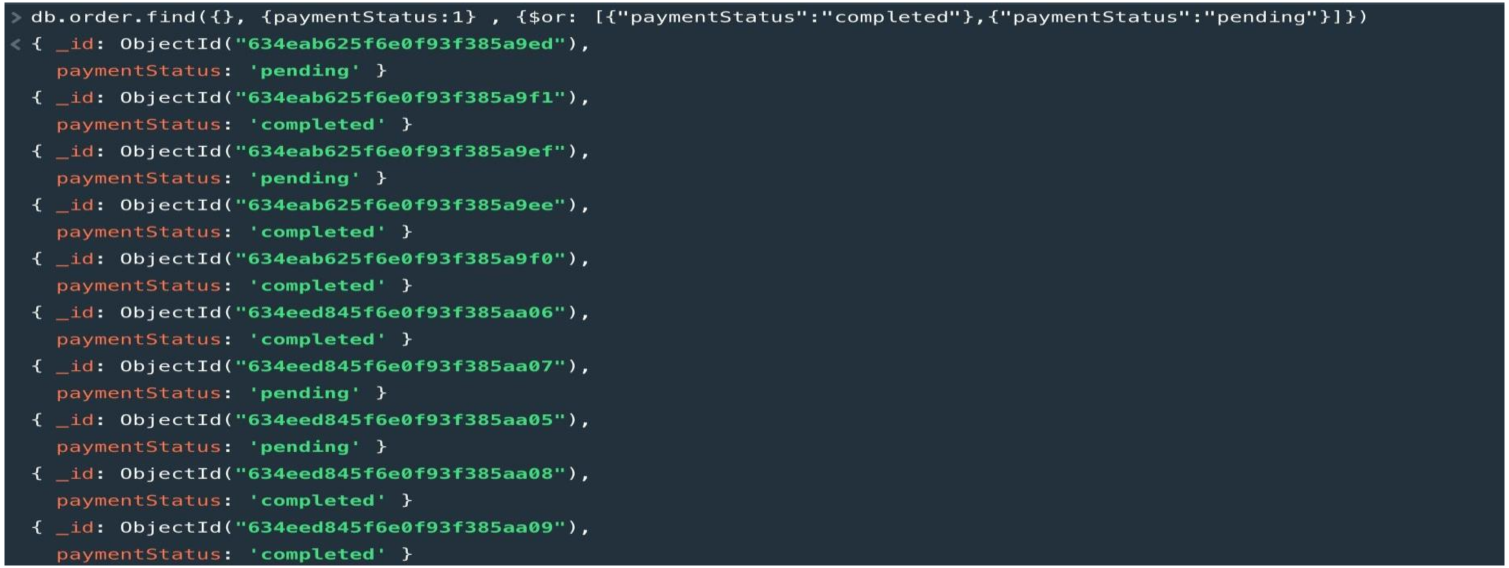


```
> db.menu.find({},{foodName:1}).sort({discount_price:-1})
< { _id: ObjectId("634eba3b5f6e0f93f385a9f3"),
    foodName: 'Butter Chicken' }
  { _id: ObjectId("634eba3b5f6e0f93f385a9f5"),
    foodName: 'Fish and  chips' }
  { _id: ObjectId("634eee295f6e0f93f385aa0d"),
    foodName: 'Fish and  chips' }
  { _id: ObjectId("634eee295f6e0f93f385aa0b"),
    foodName: 'Butter Chicken' }
  { _id: ObjectId("634eba3b5f6e0f93f385a9f4"),
    foodName: 'Pork strips' }
  { _id: ObjectId("634eee295f6e0f93f385aa0c"),
    foodName: 'Pork strips' }
  { _id: ObjectId("634eba3b5f6e0f93f385a9f6"),
    foodName: 'Mochar bora' }
  { _id: ObjectId("634eee295f6e0f93f385aa0e"),
    foodName: 'Mochar bora' }
  { _id: ObjectId("634eba3b5f6e0f93f385a9f2"),
    foodName: 'Plain rice' }
```

Business queries of NoSQL

Checking payment status of the orders

```
db.order.find({}, {paymentStatus:1} , {$or: [{"paymentStatus":"completed"},{"paymentStatus":"pending"}]})
```

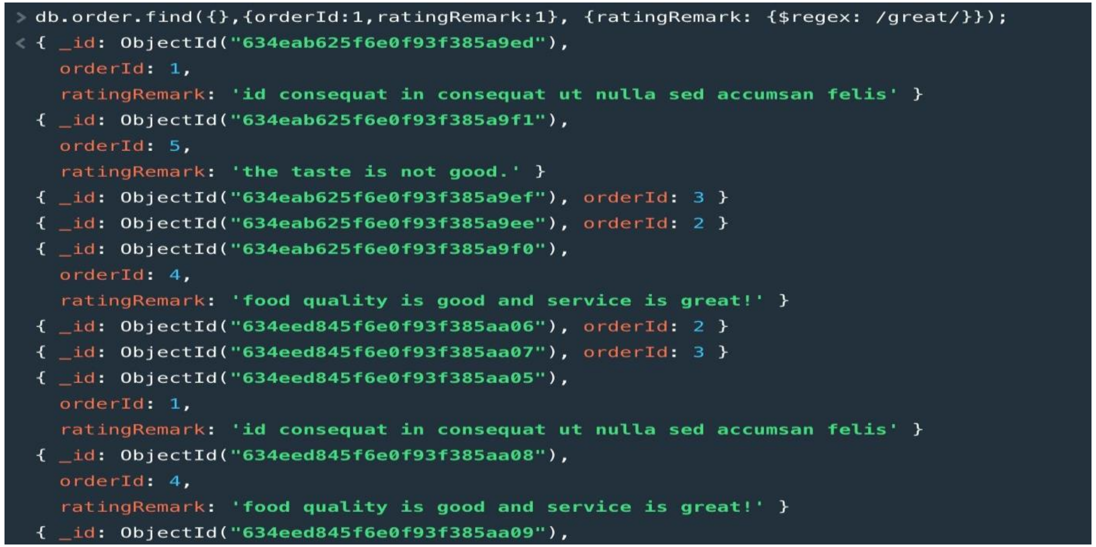


```
> db.order.find({}, {paymentStatus:1} , {$or: [{"paymentStatus":"completed"},{"paymentStatus":"pending"}]})
< { _id: ObjectId("634eab625f6e0f93f385a9ed"),
    paymentStatus: 'pending' }
  { _id: ObjectId("634eab625f6e0f93f385a9f1"),
    paymentStatus: 'completed' }
  { _id: ObjectId("634eab625f6e0f93f385a9ef"),
    paymentStatus: 'pending' }
  { _id: ObjectId("634eab625f6e0f93f385a9ee"),
    paymentStatus: 'completed' }
  { _id: ObjectId("634eab625f6e0f93f385a9f0"),
    paymentStatus: 'completed' }
  { _id: ObjectId("634eed845f6e0f93f385aa06"),
    paymentStatus: 'completed' }
  { _id: ObjectId("634eed845f6e0f93f385aa07"),
    paymentStatus: 'pending' }
  { _id: ObjectId("634eed845f6e0f93f385aa05"),
    paymentStatus: 'pending' }
  { _id: ObjectId("634eed845f6e0f93f385aa08"),
    paymentStatus: 'completed' }
  { _id: ObjectId("634eed845f6e0f93f385aa09"),
    paymentStatus: 'completed' }
```

Business queries of NoSQL

Finding rating remarks which include “great” in order to find orders which got positive feedbacks

```
db.order.find({},{orderId:1,ratingRemark:1}, {ratingRemark: {$regex: /great/}});
```

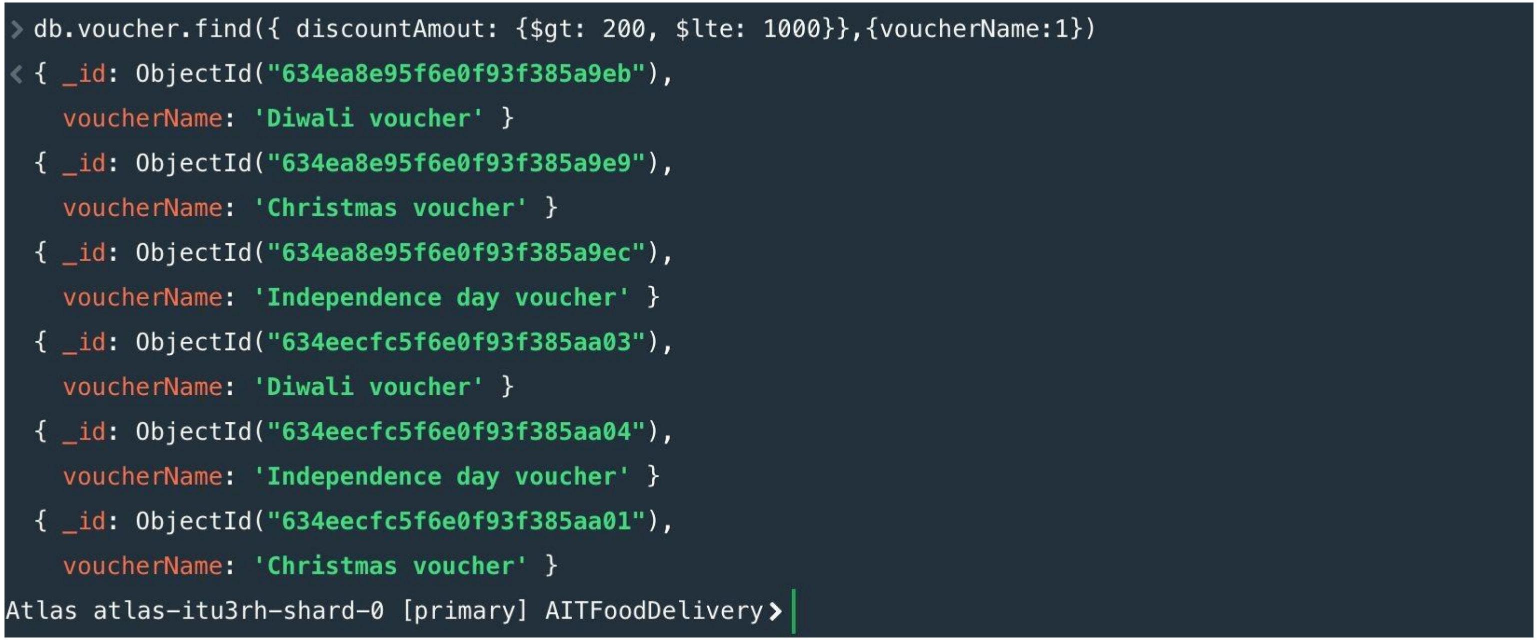


```
> db.order.find({},{orderId:1,ratingRemark:1}, {ratingRemark: {$regex: /great/}});
< { _id: ObjectId("634eab625f6e0f93f385a9ed"),
    orderId: 1,
    ratingRemark: 'id consequat in consequat ut nulla sed accumsan felis' }
  { _id: ObjectId("634eab625f6e0f93f385a9f1"),
    orderId: 5,
    ratingRemark: 'the taste is not good.' }
  { _id: ObjectId("634eab625f6e0f93f385a9ef"), orderId: 3 }
  { _id: ObjectId("634eab625f6e0f93f385a9ee"), orderId: 2 }
  { _id: ObjectId("634eab625f6e0f93f385a9f0"),
    orderId: 4,
    ratingRemark: 'food quality is good and service is great!' }
  { _id: ObjectId("634eed845f6e0f93f385aa06"), orderId: 2 }
  { _id: ObjectId("634eed845f6e0f93f385aa07"), orderId: 3 }
  { _id: ObjectId("634eed845f6e0f93f385aa05"),
    orderId: 1,
    ratingRemark: 'id consequat in consequat ut nulla sed accumsan felis' }
  { _id: ObjectId("634eed845f6e0f93f385aa08"),
    orderId: 4,
    ratingRemark: 'food quality is good and service is great!' }
  { _id: ObjectId("634eed845f6e0f93f385aa09"),
```

Business queries of NoSQL

Help customers in finding vouchers where discount amount is between 200 and 1000 THB.

```
db.voucher.find({ discountAmout: {$gt: 200, $lte: 1000}},{voucherName:1})
```

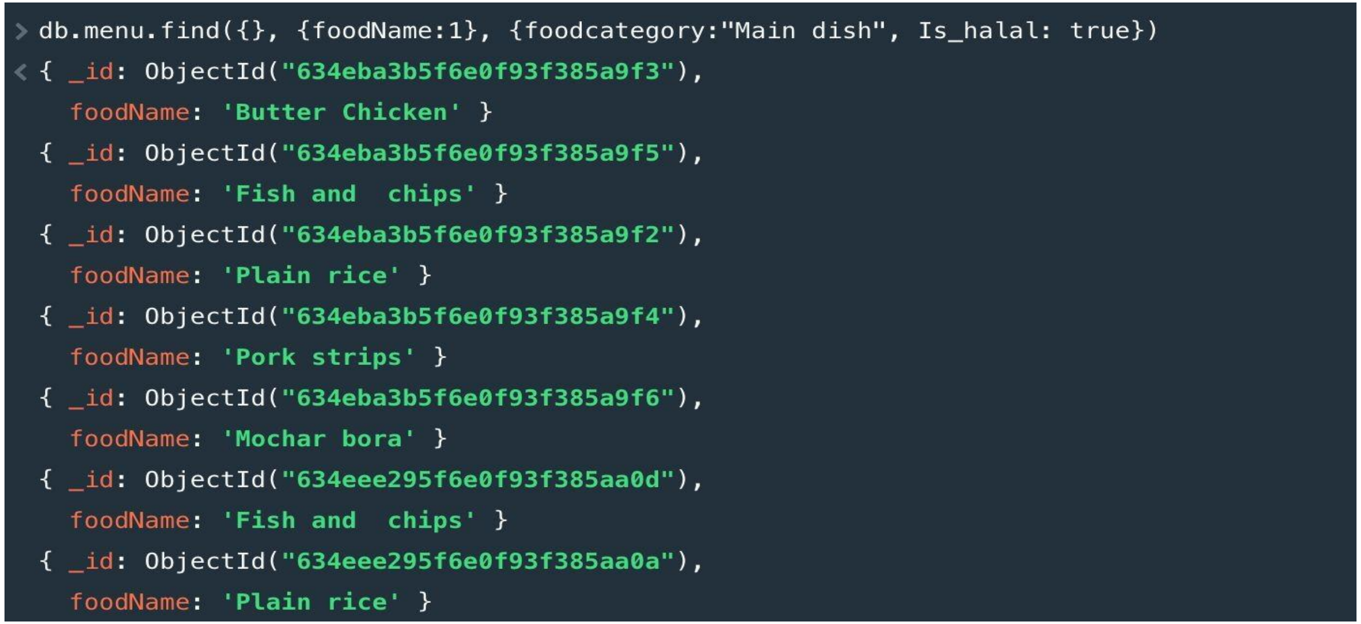


```
> db.voucher.find({ discountAmout: {$gt: 200, $lte: 1000}},{voucherName:1})
< { _id: ObjectId("634ea8e95f6e0f93f385a9eb"),
    voucherName: 'Diwali voucher' }
  { _id: ObjectId("634ea8e95f6e0f93f385a9e9"),
    voucherName: 'Christmas voucher' }
  { _id: ObjectId("634ea8e95f6e0f93f385a9ec"),
    voucherName: 'Independence day voucher' }
  { _id: ObjectId("634eecfc5f6e0f93f385aa03"),
    voucherName: 'Diwali voucher' }
  { _id: ObjectId("634eecfc5f6e0f93f385aa04"),
    voucherName: 'Independence day voucher' }
  { _id: ObjectId("634eecfc5f6e0f93f385aa01"),
    voucherName: 'Christmas voucher' }
Atlas atlas-itu3rh-shard-0 [primary] AITFoodDelivery>
```

Business queries of NoSQL

Help customers in finding halal food

```
db.menu.find({}, {foodName:1}, {foodcategory:"Main dish", Is_halal: true})
```

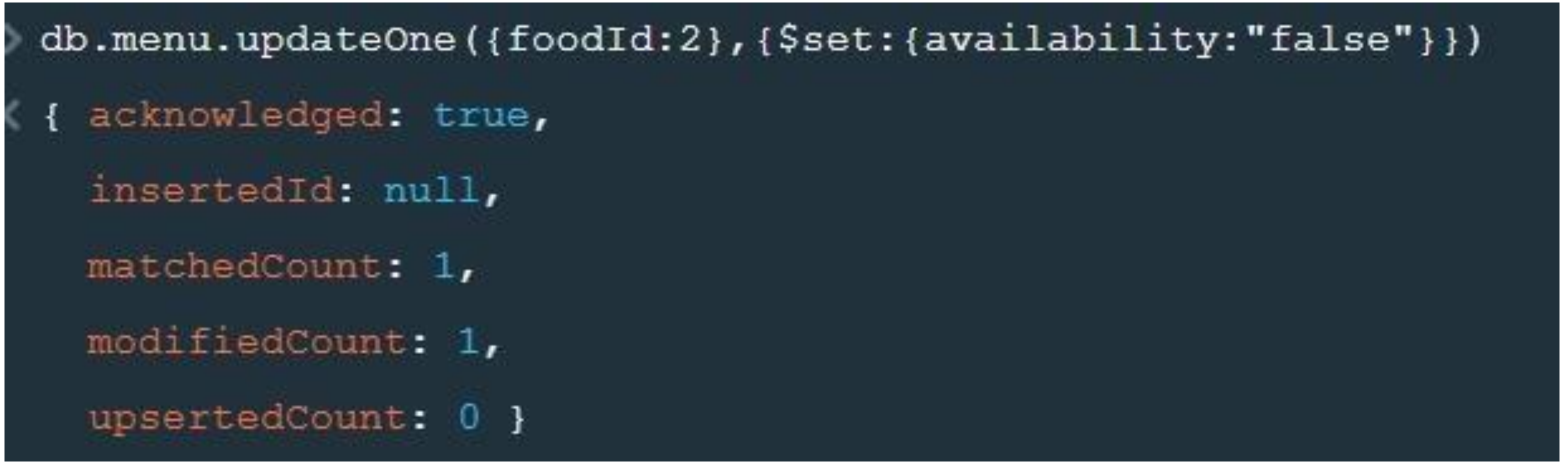


```
> db.menu.find({}, {foodName:1}, {foodcategory:"Main dish", Is_halal: true})
< { _id: ObjectId("634eba3b5f6e0f93f385a9f3"),
    foodName: 'Butter Chicken' }
  { _id: ObjectId("634eba3b5f6e0f93f385a9f5"),
    foodName: 'Fish and  chips' }
  { _id: ObjectId("634eba3b5f6e0f93f385a9f2"),
    foodName: 'Plain rice' }
  { _id: ObjectId("634eba3b5f6e0f93f385a9f4"),
    foodName: 'Pork strips' }
  { _id: ObjectId("634eba3b5f6e0f93f385a9f6"),
    foodName: 'Mochar bora' }
  { _id: ObjectId("634eee295f6e0f93f385aa0d"),
    foodName: 'Fish and  chips' }
  { _id: ObjectId("634eee295f6e0f93f385aa0a"),
    foodName: 'Plain rice' }
```

Business queries of NoSQL

Updating menu availability in menu collection

```
db.menu.updateOne({foodId:2},{$set:{availability:”false”}})
```

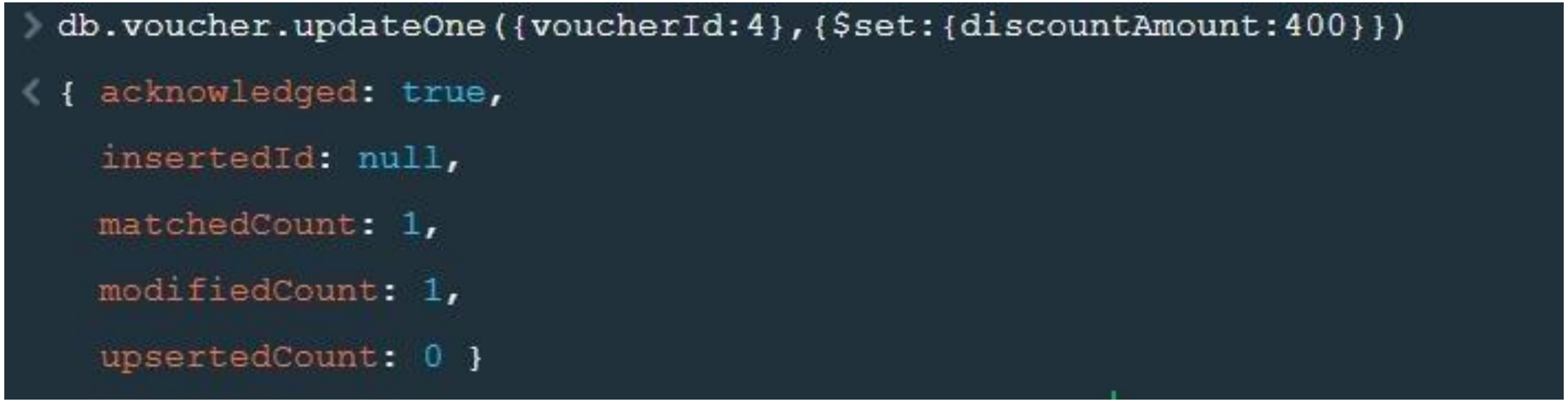


```
db.menu.updateOne({foodId:2},{$set:{availability:"false"}})
{ acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0 }
```

Business queries of NoSQL

Updating discount amount in voucher collection

```
db.voucher.updateOne({voucherId:4},{$set:{discountAmount:400}})
```

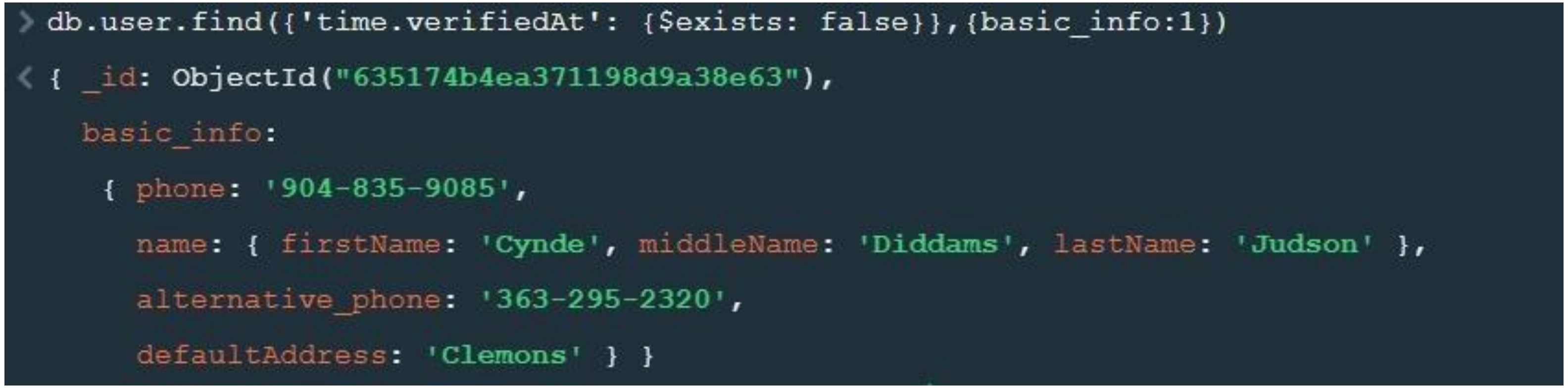


```
> db.voucher.updateOne({voucherId:4},{$set:{discountAmount:400}})
< { acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0 }
```

Business queries of NoSQL

Finding users who are registered but not verified yet

```
db.user.find({‘time.verifiedAt’: {$exists: false}},{basic_info:1})
```

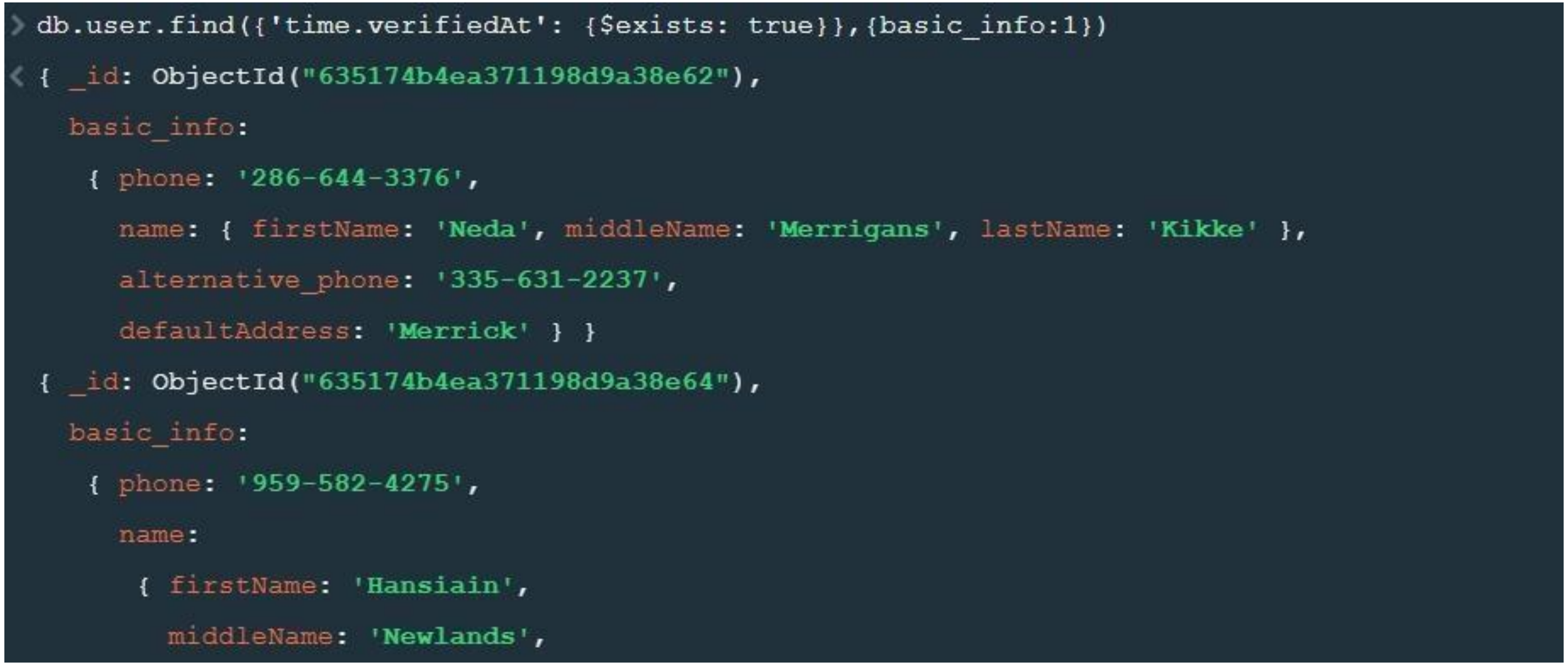


```
> db.user.find({'time.verifiedAt': {$exists: false}},{basic_info:1})
< { _id: ObjectId("635174b4ea371198d9a38e63"),
    basic_info:
     { phone: '904-835-9085',
       name: { firstName: 'Cynde', middleName: 'Diddams', lastName: 'Judson' },
       alternative_phone: '363-295-2320',
       defaultAddress: 'Clemons' } }
```

Business queries of NoSQL

Finding users whose registration and verification process is completed

```
db.user.find({‘time.verifiedAt’: {$exists: true}},{basic_info:1})
```

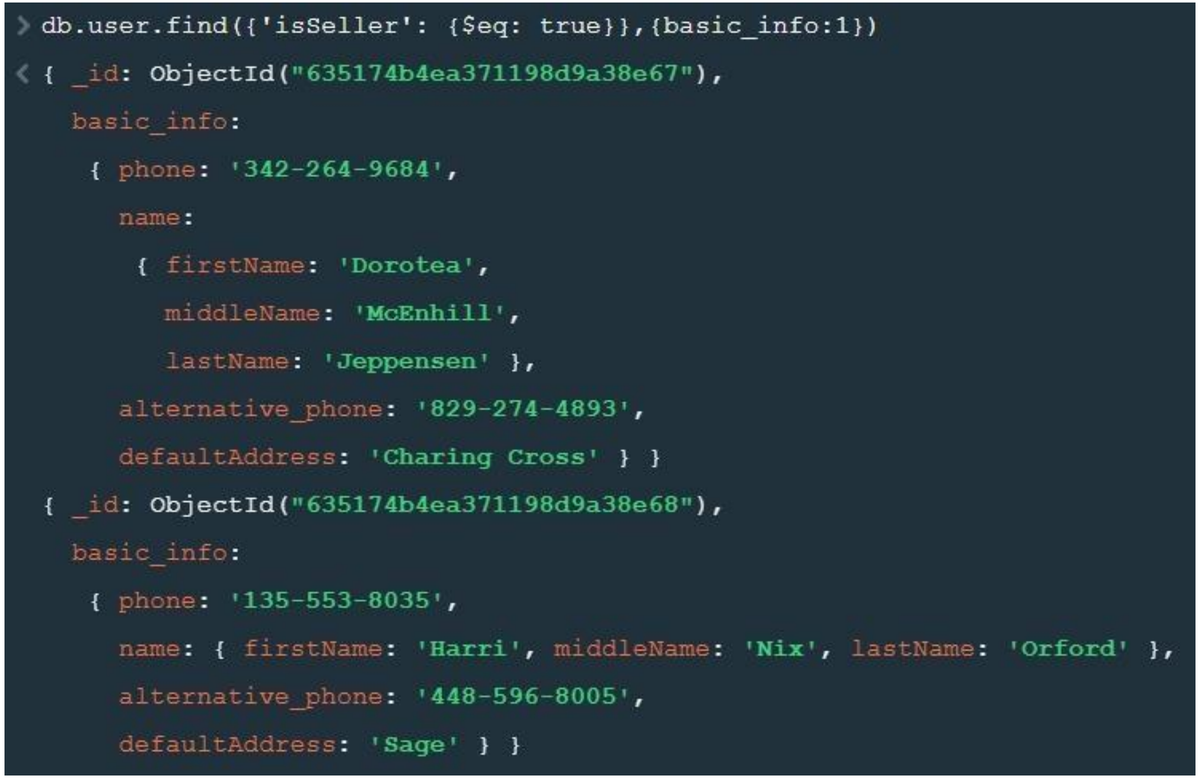


```
> db.user.find({'time.verifiedAt': {$exists: true}},{basic_info:1})
< { _id: ObjectId("635174b4ea371198d9a38e62"),
    basic_info:
     { phone: '286-644-3376',
       name: { firstName: 'Neda', middleName: 'Merrigans', lastName: 'Kikke' },
       alternative_phone: '335-631-2237',
       defaultAddress: 'Merrick' } }
  { _id: ObjectId("635174b4ea371198d9a38e64"),
    basic_info:
     { phone: '959-582-4275',
       name:
        { firstName: 'Hansiain',
          middleName: 'Newlands',
```

Business queries of NoSQL

Finding users who are registered as a buyer and as a seller

```
db.user.find({'isSeller': {$eq: true}},{basic_info:1})
```

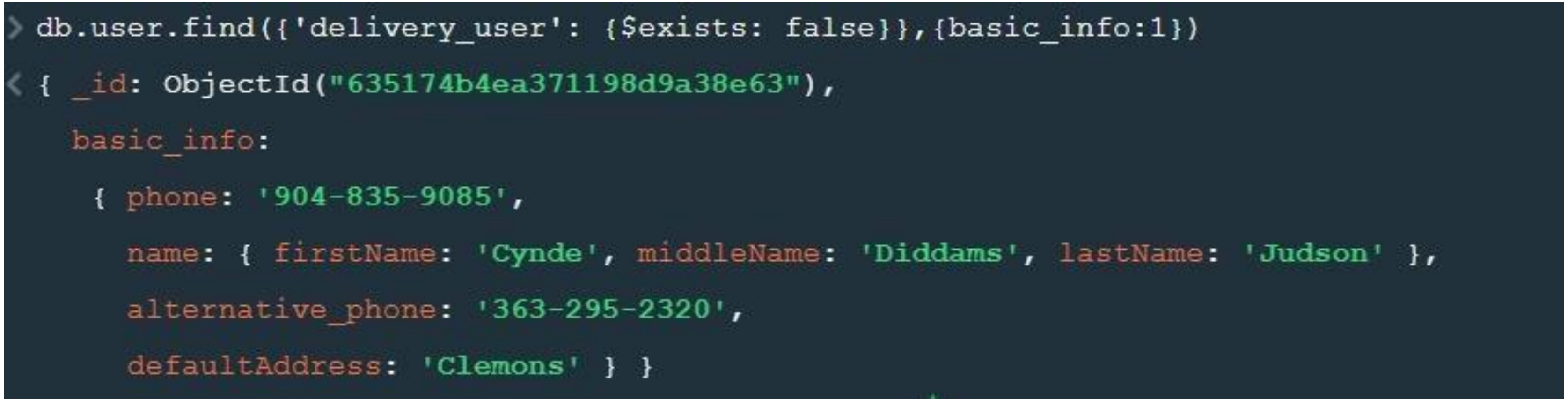


```
> db.user.find({'isSeller': {$eq: true}},{basic_info:1})
< { _id: ObjectId("635174b4ea371198d9a38e67"),
    basic_info:
     { phone: '342-264-9684',
       name:
        { firstName: 'Dorotea',
          middleName: 'McEnhill',
          lastName: 'Jeppensen' },
       alternative_phone: '829-274-4893',
       defaultAddress: 'Charing Cross' } }
  { _id: ObjectId("635174b4ea371198d9a38e68"),
    basic_info:
     { phone: '135-553-8035',
       name: { firstName: 'Harri', middleName: 'Nix', lastName: 'Orford' },
       alternative_phone: '448-596-8005',
       defaultAddress: 'Sage' } }
```

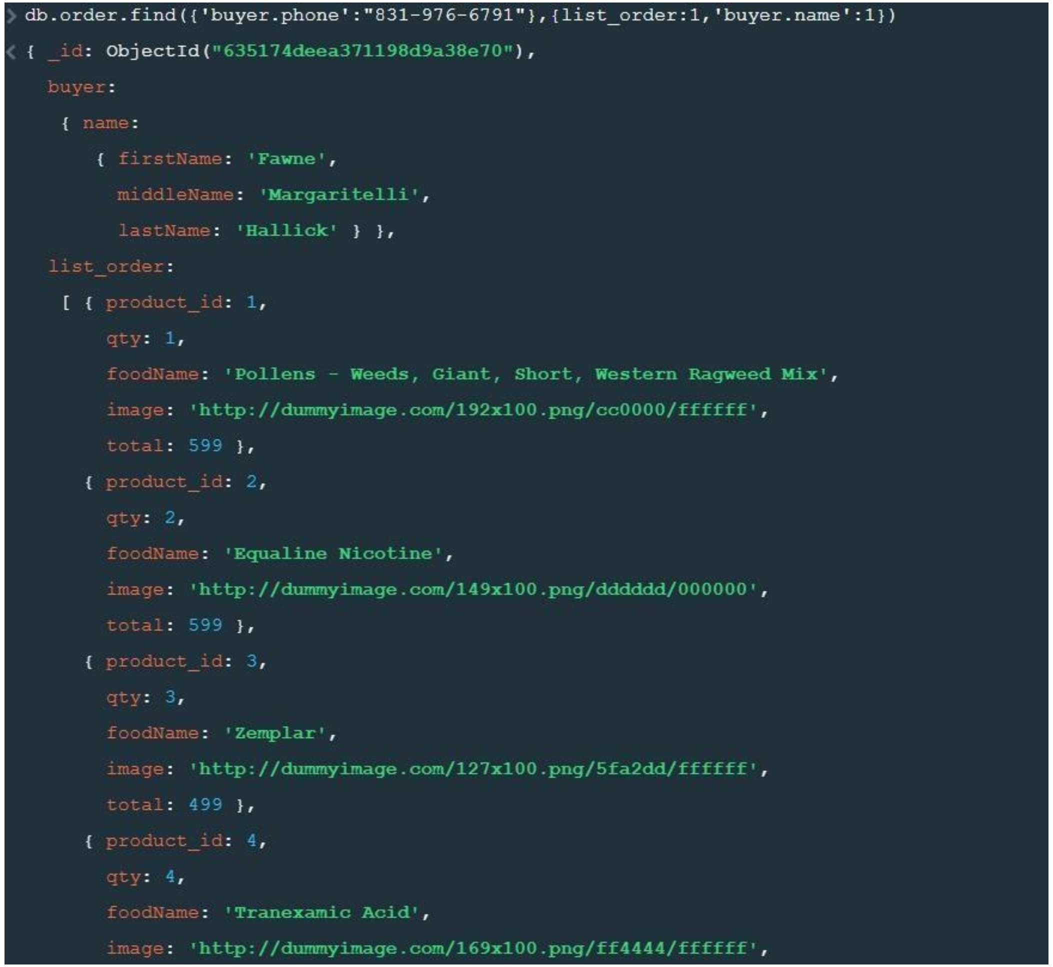
Business queries of NoSQL

Finding users who are not doing delivery

```
db.user.find({‘delivery_user’: {$exists: false}},{basic_info:1})
```



```
> db.user.find({'delivery_user': {$exists: false}},{basic_info:1})
< { _id: ObjectId("635174b4ea371198d9a38e63"),
    basic_info:
     { phone: '904-835-9085',
       name: { firstName: 'Cynde', middleName: 'Diddams', lastName: 'Judson' },
       alternative_phone: '363-295-2320',
       defaultAddress: 'Clemons' } }
```

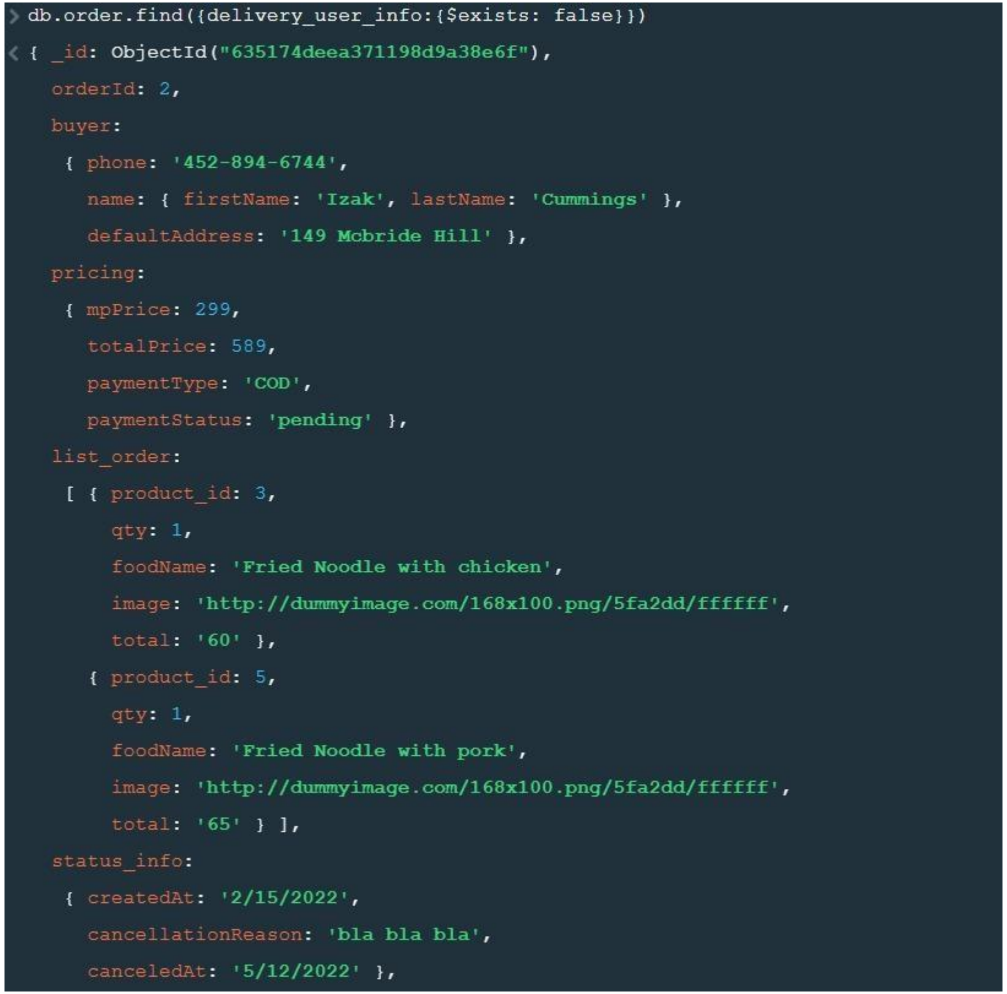


```
> db.order.find({'buyer.phone':"831-976-6791"},{list_order:1,'buyer.name':1})
< { _id: ObjectId("635174deea371198d9a38e70"),
    buyer:
     { name:
        { firstName: 'Fawne',
          middleName: 'Margaritelli',
          lastName: 'Hallick' } },
    list_order:
     [ { product_id: 1,
         qty: 1,
         foodName: 'Pollens - Weeds, Giant, Short, Western Ragweed Mix',
         image: 'http://dummyimage.com/192x100.png/cc0000/ffffff',
         total: 599 },
       { product_id: 2,
         qty: 2,
         foodName: 'Equaline Nicotine',
         image: 'http://dummyimage.com/149x100.png/dddddd/000000',
         total: 599 },
       { product_id: 3,
         qty: 3,
         foodName: 'Zemplar',
         image: 'http://dummyimage.com/127x100.png/5fa2dd/ffffff',
         total: 499 },
       { product_id: 4,
         qty: 4,
         foodName: 'Tranexamic Acid',
         image: 'http://dummyimage.com/169x100.png/ff4444/ffffff',
```

Business queries OfNoSQL

User is checking his order list

```
db.user.find({‘buyer.phone’
:”831-976-6791”},{list_orde
r:1, ‘buyer.name’:1})
```



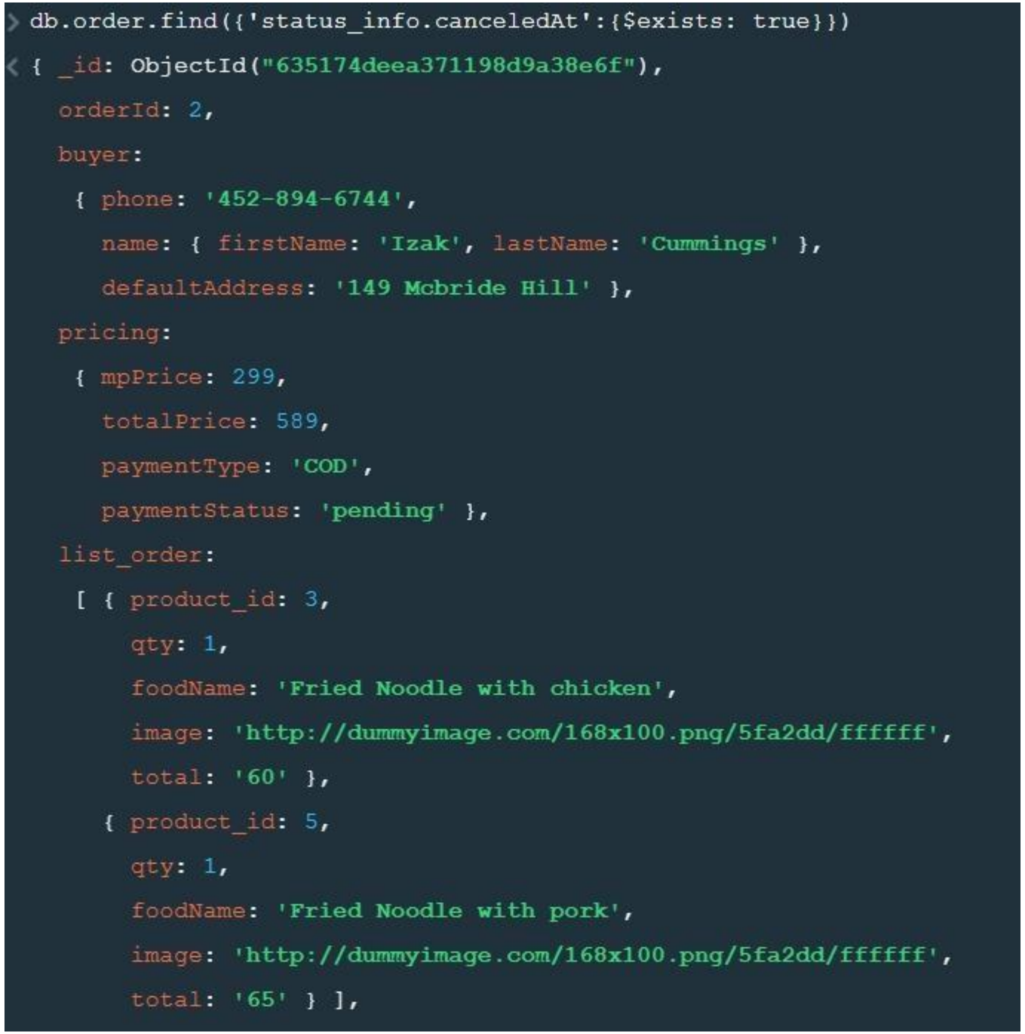
```
> db.order.find({delivery_user_info:{$exists: false}})
< { _id: ObjectId("635174deea371198d9a38e6f"),
    orderId: 2,
    buyer:
     { phone: '452-894-6744',
       name: { firstName: 'Izak', lastName: 'Cummings' },
       defaultAddress: '149 Mcbride Hill' },
    pricing:
     { mpPrice: 299,
       totalPrice: 589,
       paymentType: 'COD',
       paymentStatus: 'pending' },
    list_order:
     [ { product_id: 3,
         qty: 1,
         foodName: 'Fried Noodle with chicken',
         image: 'http://dummyimage.com/168x100.png/5fa2dd/ffffff',
         total: '60' },
       { product_id: 5,
         qty: 1,
         foodName: 'Fried Noodle with pork',
         image: 'http://dummyimage.com/168x100.png/5fa2dd/ffffff',
         total: '65' } ],
    status_info:
     { createdAt: '2/15/2022',
       cancellationReason: 'bla bla bla',
       canceledAt: '5/12/2022' },
```

Business queries of NoSQL

Finding users who will pick up the order(s) by themselves

```
db.order.find({delivery_
user_info:{$exists:
false}})
```

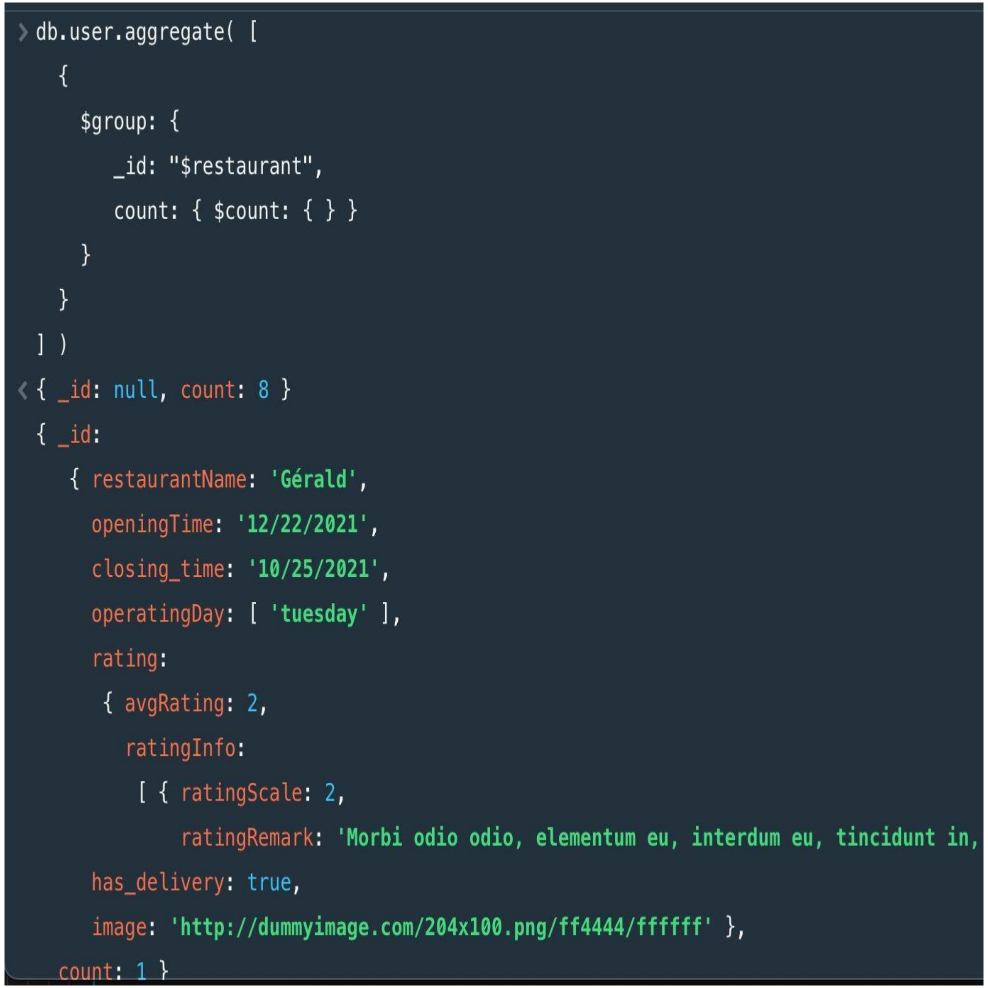
Business queries of NoSQL


```
> db.order.find({'status_info.canceledAt':{$exists: true}})
< { _id: ObjectId("635174deea371198d9a38e6f"),
    orderId: 2,
    buyer:
     { phone: '452-894-6744',
       name: { firstName: 'Izak', lastName: 'Cummings' },
       defaultAddress: '149 Mcbride Hill' },
    pricing:
     { mpPrice: 299,
       totalPrice: 589,
       paymentType: 'COD',
       paymentStatus: 'pending' },
    list_order:
     [ { product_id: 3,
         qty: 1,
         foodName: 'Fried Noodle with chicken',
         image: 'http://dummyimage.com/168x100.png/5fa2dd/ffffff',
         total: '60' },
       { product_id: 5,
         qty: 1,
         foodName: 'Fried Noodle with pork',
         image: 'http://dummyimage.com/168x100.png/5fa2dd/ffffff',
         total: '65' } ],
```

Finding user(s) who cancelled the order(s)

```
db.order.find({‘status_info
.canceledAt’:{$exists:
true}})
```

Business queries of NoSQL



```
> db.user.aggregate( [
    {
      $group: {
         _id: "$restaurant",
         count: { $count: { } }
      }
    }
  ] )
< { _id: null, count: 8 }
  { _id:
     { restaurantName: 'Gérald',
       openingTime: '12/22/2021',
       closing_time: '10/25/2021',
       operatingDay: [ 'tuesday' ],
       rating:
        { avgRating: 2,
          ratingInfo:
           [ { ratingScale: 2,
               ratingRemark: 'Morbi odio odio, elementum eu, interdum eu, tincidunt in,
       has_delivery: true,
       image: 'http://dummyimage.com/204x100.png/ff4444/ffffff' },
    count: 1 }
```

Finding user(s) as restaurants and count it from all of users

```
db.user.aggregate( [
    {
      $group: {
         _id: "$restaurant",
         count: { $count: { } }
      }
    }
] )
```

Thank You!