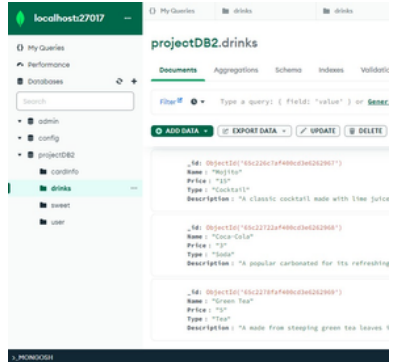


# Query Demonstrations:

## Update documents in drinks collection

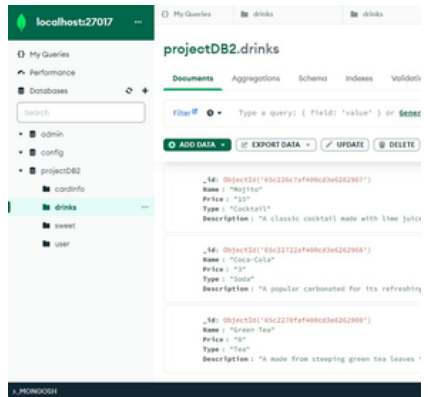
### Updated documents on the price of green tea

```
mongosh mongodb://127.0.0.1:27017?directConnection=true&serverSelectionTimeoutMS=2000
> use projectDB2
> db.drinks.find({})
{
  "_id": ObjectId("65c221a8f460dc3e422965"),
  "Name": "Caramel Macchiato",
  "Price": "28",
  "Type": "Coffee",
  "Description": "A layered espresso beverage made with steamed milk, vanilla syrup, espresso."
},
{
  "_id": ObjectId("65c228c7f460dc3e422968"),
  "Name": "Orange Juice",
  "Price": "18",
  "Type": "Juice",
  "Description": "A natural beverage made from freshly squeezed oranges."
},
{
  "_id": ObjectId("65c228c7f460dc3e422967"),
  "Name": "Mojito",
  "Price": "15",
  "Type": "Cocktail",
  "Description": "A classic cocktail made with lime juice, soda water, and mint leaves."
},
{
  "_id": ObjectId("65c22722af460dc3e422968"),
  "Name": "Coca-Cola",
  "Price": "3",
  "Type": "Soda",
  "Description": "A popular carbonated for its refreshing taste."
},
{
  "_id": ObjectId("65c2278faf460dc3e422969"),
  "Name": "Green Tea",
  "Price": "5",
  "Type": "Tea",
  "Description": "A made from steeping green tea leaves in hot water."
}
{
  "updatedCount": 0,
  "projectDB2.drinks.updateOne($set: {Name: 'Green Tea'}, {$set: {Price: '10'}}),
  "acknowledged": true,
  "insertedId": null,
  "matchedCount": 1,
  "modifiedCount": 1
}
```



### Its price has been changed from 5 to 8

```
projectDB2> db.drinks.find({})
{
  "_id": ObjectId("65c221a8f460dc3e422965"),
  "Name": "Caramel Macchiato",
  "Price": "28",
  "Type": "Coffee",
  "Description": "A layered espresso beverage made with steamed milk, vanilla syrup, espresso."
},
{
  "_id": ObjectId("65c228c7f460dc3e422968"),
  "Name": "Orange Juice",
  "Price": "18",
  "Type": "Juice",
  "Description": "A natural beverage made from freshly squeezed oranges."
},
{
  "_id": ObjectId("65c228c7f460dc3e422967"),
  "Name": "Mojito",
  "Price": "15",
  "Type": "Cocktail",
  "Description": "A classic cocktail made with lime juice, soda water, and mint leaves."
},
{
  "_id": ObjectId("65c22722af460dc3e422968"),
  "Name": "Coca-Cola",
  "Price": "3",
  "Type": "Soda",
  "Description": "A popular carbonated for its refreshing taste."
},
{
  "_id": ObjectId("65c2278faf460dc3e422969"),
  "Name": "Green Tea",
  "Price": "8",
  "Type": "Tea",
  "Description": "A made from steeping green tea leaves in hot water."
}
}
```



# Query Demonstrations:

## Update documents in cardInfo collection

### Updated documents on the expirationDate of Shahad AlZahrani

```
mongosh mongo@127.0.0.1:27017/admin?connect=true&serverSelectionTimeoutMS=2000
> use projectDB2
> db.cardInfo.updateOne(
  {
    '_id': ObjectId('65c21a093f095cab04ed7726'),
    'cardNumber': '9999999911112222',
    'cardType': 'Visa',
    'cardHolderName': 'RanaAlqarni',
    'CVC': '565',
    'expirationDate': '09/27'
  },
  {
    '$set': {
      'expirationDate': '12/29'
    }
  },
  {
    'upserted': 0
  }
)
```

projectDB2> db.cardInfo.updateOne( {CVC:'999'}, { \$set: { expirationDate: '12/29' } } )

localhost:27017

My Queries

Performance

Databases

Search

Filter

ADD DATA

EXPORT DATA

UPDATE

DELETE

Documents

Aggregations

Schema

Indexes

Validation

cardInfo

expirationDate: "12/29"

cardInfo

cardNumber: "9999999911112222"

cardType: "Visa"

cardHolderName: "RanaAlqarni"

CVC: "565"

expirationDate: "09/27"

cardInfo

cardNumber: "9999345011118022"

cardType: "Visa"

cardHolderName: "ShahadAlzahrani"

CVC: "999"

expirationDate: "12/29"

cardInfo

cardNumber: "876543210987654"

cardType: "Mastercard"

cardHolderName: "RanaAlqarni"

CVC: "199"

expirationDate: "01/29"

cardInfo

cardNumber: "876543210987654"

cardType: "Mastercard"

cardHolderName: "ShahadAlzahrani"

CVC: "999"

expirationDate: "12/29"

cardInfo

cardNumber: "876543210987654"

cardType: "Mastercard"

cardHolderName: "RanaAlqarni"

CVC: "199"

expirationDate: "01/29"

### The expirationDate of the card has been changed from 12/24 to 12/29

```
cardInfo: {
  expirationDate: "12/29"
}
cardInfo: {
  _id: ObjectId('65c21a093f095cab04ed7726'),
  cardNumber: '876543210987654',
  cardType: 'Mastercard',
  cardHolderName: 'FatimaAlmohd',
  CVC: '789',
  expirationDate: '08/28'
}
cardInfo: {
  _id: ObjectId('65c21a093f095cab04ed7726'),
  cardNumber: '1111222233334444',
  cardType: 'Visa',
  cardHolderName: 'RanaAlqarni',
  CVC: '999',
  expirationDate: '08/28'
}
cardInfo: {
  _id: ObjectId('65c21a093f095cab04ed7726'),
  cardNumber: '9999999911112222',
  cardType: 'Visa',
  cardHolderName: 'RanaAlqarni',
  CVC: '565',
  expirationDate: '09/27'
}
cardInfo: {
  _id: ObjectId('65c21a093f095cab04ed7726'),
  cardNumber: '7777888899990000',
  cardType: 'Visa',
  cardHolderName: 'RanaAlqarni',
  CVC: '565',
  expirationDate: '11/25'
}
cardInfo: {
  _id: ObjectId('65c21a093f095cab04ed7726'),
  cardNumber: '9999345011118022',
  cardType: 'Visa',
  cardHolderName: 'ShahadAlzahrani',
  CVC: '999',
  expirationDate: '12/29'
}
```

localhost:27017

My Queries

Performance

Databases

Search

Filter

ADD DATA

EXPORT DATA

UPDATE

DELETE

Documents

Aggregations

Schema

Indexes

Validation

cardInfo

cardNumber: "876543210987654"

cardType: "Mastercard"

cardHolderName: "RanaAlqarni"

CVC: "789"

expirationDate: "08/28"

cardInfo

cardNumber: "1111222233334444"

cardType: "Visa"

cardHolderName: "RanaAlqarni"

CVC: "999"

expirationDate: "08/28"

cardInfo

cardNumber: "9999999911112222"

cardType: "Visa"

cardHolderName: "RanaAlqarni"

CVC: "565"

expirationDate: "09/27"

cardInfo

cardNumber: "7777888899990000"

cardType: "Visa"

cardHolderName: "RanaAlqarni"

CVC: "565"

expirationDate: "11/25"

cardInfo

cardNumber: "9999345011118022"

cardType: "Visa"

cardHolderName: "ShahadAlzahrani"

CVC: "999"

expirationDate: "12/29"

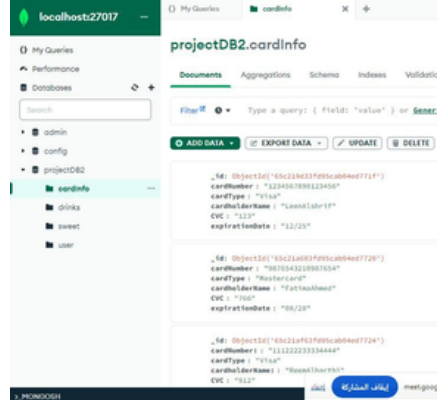
# Query Demonstrations:

## Delete document from cardInfo collection

Here card of Fatima Ahmed is present

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
projectDB2> db.cardInfo.find({})

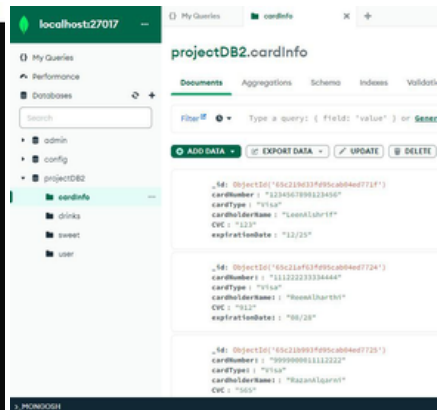
[
  {
    _id: ObjectId('65c219d33fd95cab04ed771f'),
    cardNumber: '1234567890123456',
    cardType: 'Visa',
    cardholderName: 'LeenAlsharif',
    CVC: '123',
    expirationDate: '12/25'
  },
  {
    _id: ObjectId('65c21a603fd95cab04ed7720'),
    cardNumber: '9876543210987654',
    cardType: 'Mastercard',
    cardholderName: 'FatimaAhmed',
    CVC: '766',
    expirationDate: '08/28'
  },
  {
    _id: ObjectId('65c21af63fd95cab04ed7724'),
    cardNumber: '11122233334444',
    cardType: 'Visa',
    cardholderName: 'ReemAlharthi',
    CVC: '912',
    expirationDate: '08/28'
  }
]
```



## Delete card of Fatima Ahmed

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
projectDB2> db.cardInfo.deleteOne({CVC:'766'})
{ acknowledged: true, deletedCount: 1 }
projectDB2> db.cardInfo.find({})

[
  {
    _id: ObjectId('65c219d33fd95cab04ed771f'),
    cardNumber: '1234567890123456',
    cardType: 'Visa',
    cardholderName: 'LeenAlsharif',
    CVC: '123',
    expirationDate: '12/25'
  },
  {
    _id: ObjectId('65c21af63fd95cab04ed7724'),
    cardNumber: '11122233334444',
    cardType: 'Visa',
    cardholderName: 'ReemAlharthi',
    CVC: '912',
    expirationDate: '08/28'
  },
  {
    _id: ObjectId('65c21b993fd95cab04ed7725'),
    cardNumber: '9999000011112222',
    cardType: 'Visa',
    cardholderName: 'RazanAlqarni',
    CVC: '565',
    expirationDate: '09/27'
  }
]
```



# Query Demonstrations:

insert new user to User collection

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
```

```
{
  "_id": ObjectId("65c213f93fd95cab04ed7713"),
  "username": "Rola3",
  "password": "rola31",
  "phoneNumber": "96655898562",
  "email": "rola@hotmail.com"
}
projectDB2> db.user.insertOne({username:"hanan", password:"123456", phoneNumber:"966435675678", email:"hanan@gmail.com"})
```

The screenshot shows the MongoDB Compass application running on localhost:27017. The left sidebar displays a tree view of databases and collections. The 'projectDB2' database is expanded, showing collections: 'cardInfo', 'drinks', 'sweet', and 'user'. The 'user' collection is selected and highlighted. The top toolbar shows 'My Queries' and 'user' tabs. The main area displays the 'projectDB2.user' collection with tabs for 'Documents', 'Aggregations', 'Schema', 'Indexes', and 'Validation'. The 'Documents' tab is active, showing a list of documents. A 'Filter' bar is present with the text 'Type a query: { field: 'value' } or [Generate](#)'. Below the filter are buttons for 'ADD DATA', 'EXPORT DATA', 'UPDATE', and 'DELETE'. Three documents are displayed in a list view, each showing its JSON structure. The third document is the newly added user 'hanan'.

```
{ "_id": ObjectId("65c213843fd95cab04ed7712"), "username": "shahad2", "password": "sh0006", "phoneNumber": "966588931274", "email": "shahad@hotmail.com" }
```

```
{ "_id": ObjectId("65c213f93fd95cab04ed7713"), "username": "Rola3", "password": "rola31", "phoneNumber": "96655898562", "email": "rola@hotmail.com" }
```

```
{ "_id": ObjectId("65c25785a48272c2304a4462"), "username": "hanan", "password": "123456", "phoneNumber": "966435675678", "email": "hanan@gmail.com" }
```

> MONGOSH

Adding a new user of username hanan

# Query Demonstrations:

## Retrieve the distinct card type in cardInfo collection

```
mongosh mongodb://127.0.0.1:27017/?directConnections=true&serverSelectionTimeoutMS=2000
{
  'cardholderName': 'ReemAlharthi',
  'CVC': '912',
  'expirationDate': '08/28'
},
{
  '_id': ObjectId('65c21b993fd95cab04ed7725'),
  'cardNumber': '9999080011112222',
  'cardType': 'Visa',
  'cardholderName': 'RazanAlqarni',
  'CVC': '565',
  'expirationDate': '09/27'
},
{
  '_id': ObjectId('65c21c443fd95cab04ed7726'),
  'cardNumber': '7777666655554444',
  'cardType': 'Visa',
  'cardholderName': 'RanaJaha',
  'CVC': '377',
  'expirationDate': '11/25'
},
{
  '_id': ObjectId('65c21ca43fd95cab04ed7727'),
  'cardNumber': '9999345011110022',
  'cardType': 'Visa',
  'cardholderName': 'ShahadAlzhrani',
  'CVC': '999',
  'expirationDate': '12/29'
},
{
  '_id': ObjectId('65c21d293fd95cab04ed7728'),
  'cardNumber': '8765543210987694',
  'cardType': 'Mastercard',
  'cardholderName': 'RolaAlotibi',
  'CVC': '199',
  'expirationDate': '01/29'
}
]
project082> db.cardInfo.distinct("cardType")
[ 'Visa' ]
```

# Query Demonstrations:

Retrieves only the first two documents from the sweet collection

```
projectDB2> db.sweet.find().limit(2)
{
  "_id": ObjectId("65c227f7af480cd3e6295b"),
  "name": "Chocolate Delight",
  "price": "15",
  "type": "Cupcakes",
  "description": "Indulge in the rich and moist Chocolate Delight cupcakes. These heaven..",
},
{
  "_id": ObjectId("65c228a9af480cd3e6295d"),
  "name": "Classic Chocolate Chip",
  "price": "7",
  "type": "Cookies",
  "description": "These soft and chewy cookies are loaded with gooey chocolate chips and..",
}
projectDB2>
```

The screenshot shows the MongoDB Compass application running on localhost:27017. The left sidebar displays a tree view of databases and collections, with 'projectDB2' expanded to show the 'sweet' collection. The main panel is titled 'projectDB2.sweet' and has tabs for 'Documents', 'Aggregations', 'Schema', 'Indexes', and 'Validation'. The 'Documents' tab is active, showing a list of documents. A filter bar at the top of the document list allows for filtering by field or generating a query. Below the filter bar are buttons for 'ADD DATA', 'EXPORT DATA', 'UPDATE', and 'DELETE'. The document list contains two entries, each with a truncated JSON representation of a document from the 'sweet' collection.

Document
<pre>{   "_id": ObjectId("65c227f7af480cd3e6295b"),   "name": "Chocolate Delight",   "price": "15",   "type": "Cupcakes",   "description": "Indulge in the rich and moist Chocolate Delight cupcakes. These heaven..", }</pre>
<pre>{   "_id": ObjectId("65c228a9af480cd3e6295d"),   "name": "Classic Chocolate Chip",   "price": "7",   "type": "Cookies",   "description": "These soft and chewy cookies are loaded with gooey chocolate chips and..", }</pre>

# Query Demonstrations:

Retrieves all documents from the user collection, and sorts them in ascending order based on the "email" field

mongosh mongodb://127.0.0.1:27017/?directConnections=true&serverSelectionTimeoutMS=2000

```
projectDB2> db.user.find().sort({email:1})
[
  {
    _id: ObjectId('65c25785a48272c2304a4462'),
    username: 'hanan',
    password: '123456',
    phoneNumber: '966435675678',
    email: 'hanan@gmail.com'
  },
  {
    _id: ObjectId('65c2116d3fd95cab04ed770e'),
    username: 'lona',
    password: '123456',
    phoneNumber: '966555679932',
    email: 'lo@gmail.co'
  },
  {
    _id: ObjectId('65c212a53fd95cab04ed7711'),
    username: 'rana99',
    password: 'ran987',
    phoneNumber: '966533127874',
    email: 'rana1@gmail.com'
  },
  {
    _id: ObjectId('65c212423fd95cab04ed7710'),
    username: 'razan',
    password: '1a2z',
    phoneNumber: '96655349812',
    email: 'raz@gmail.com'
  },
  {
    _id: ObjectId('65c211e33fd95cab04ed770f'),
    username: 'reem',
    password: 'reem12',

```

mongosh mongodb://127.0.0.1:27017/?directConnections=true&serverSelectionTimeoutMS=2000

```
{
  _id: ObjectId('65c211e33fd95cab04ed770f'),
  username: 'reem',
  password: 'reem12',
  phoneNumber: '966555037893',
  email: 'reem@hotmail.com'
},
{
  _id: ObjectId('65c213f93fd95cab04ed7713'),
  username: 'Rola3',
  password: 'rola31',
  phoneNumber: '966555098562',
  email: 'rola@hotmail.com'
},
{
  _id: ObjectId('65c213843fd95cab04ed7712'),
  username: 'shahadZ',
  password: 'sh0000',
  phoneNumber: '966588931274',
  email: 'shahad@hotmail.com'
},
{
  _id: ObjectId('65c211063fd95cab04ed770d'),
  username: 'tima',
  password: '102935',
  phoneNumber: '966555837123',
  email: 'tima@gmail.com'
}
]
projectDB2>
```