

Mobilization of Products Via Amazon Logistics

Milestone: Implementation in NoSQL

Group 6

Ratna Manjeera Grandhi
Phani Bhavana Atluri

617-971-6775 (Telephone of Student 1)
513-206-2263 (Telephone of Student 2)

grandhi.r@northeastern.edu
atluri.p@northeastern.edu

Percentage of Effort Contributed by Student1: _50%_

Percentage of Effort Contributed by Student2: _50%_

Signature of Student 1: Ratna Manjeera Grandhi

Signature of Student 2: Phani Bhavana Atluri

Submission Date: 12/03/2022

1. Create a Sample Database schema with using a insert statement:

User Table:

```
db.users.insert({
  "user_id":1234,
  "user_name":"AMAZON LLC",
  "user_type":"BUSINESS",
  "active":"Yes",
  "subscription":"Yearly",
  "attribute1":"Online",
  "attribute2":"",
  "attribute3":""
});
```

```
db.users.insert({
  "user_id":2345,
  "user_name":"LINCOLN COLN",
  "user_type":"INDIVIDUAL",
  "active":"Yes",
  "subscription":"Quarterly",
  "attribute1":"Both",
  "attribute2":"",
  "attribute3":""
});
```

```
db.users.insert({
  "user_id":3456,
  "user_name":"SKIPPER LLC",
  "user_type":"BUSINESS",
  "active":"No",
  "subscription":"Half Yearly",
  "attribute1":"Online",
  "attribute2":"",
  "attribute3":""
});
```

```
db.users.insert({
  "user_id":4567,
  "user_name":"ZINC INC",
  "user_type":"BUSINESS",
  "active":"Yes",
  "subscription":"Yearly",
  "attribute1":"Store",
```

```
    "attribute2":"","  
    "attribute3":"","  
});
```

```
db.users.insert({  
    "user_id":5678,  
    "user_name":"ZOMBIE LLC",  
    "user_type":"BUSINESS",  
    "active":"Yes",  
    "subscription":"Yearly",  
    "attribute1":"Both",  
    "attribute2":"","  
    "attribute3":"","  
});
```

```
db.users.insert({  
    "user_id":6789,  
    "user_name":"DOOR BUSS INC",  
    "user_type":"BUSINESS",  
    "active":"Yes",  
    "subscription":"Half Yearly",  
    "attribute1":"Online",  
    "attribute2":"","  
    "attribute3":"","  
});
```

```
db.users.insert({  
    "user_id":7890,  
    "user_name":"AMAZON INC",  
    "user_type":"BUSINESS",  
    "active":"Yes",  
    "subscription":"Quarterly",  
    "attribute1":"Store",  
    "attribute2":"","  
    "attribute3":"","  
});
```

```
db.users.insert({  
    "user_id":8901,  
    "user_name":"CISCO LLC",  
    "user_type":"BUSINESS",  
    "active":"Yes",  
    "subscription":"Yearly",  
    "attribute1":"Online",  
    "attribute2":"","  
    "attribute3":"","
```

```
});
```

```
db.users.insert({
  "user_id":9012,
  "user_name":"BEXLEY APARTMENTS",
  "user_type":"BUSINESS",
  "active":"Yes",
  "subscription":"Yearly",
  "attribute1":"HOUSING",
  "attribute2":"","
  "attribute3":""
});
```

```
db.users.insert({
  "user_id":11234,
  "user_name":"UHAUL LLC",
  "user_type":"BUSINESS",
  "active":"Yes",
  "subscription":"Half Yearly",
  "attribute1":"Online",
  "attribute2":"Moving",
  "attribute3":""
});
```

```
db.users.find();
```

Using: wine (mongo)

Write your statement below and press "Run" to see the result.

```
90  "user_id":9012,
91  "user_name":"BEXLEY APARTMENTS",
92  "user_type":"BUSINESS",
93  "active":"Yes",
94  "subscription":"Yearly",
95  "attribute1":"HOUSING",
96  "attribute2":"","
97  "attribute3":""
98  });
99
100 db.users.insert({
101   "user_id":11234,
102   "user_name":"UHAUL LLC",
103   "user_type":"BUSINESS",
104   "active":"Yes",
105   "subscription":"Half Yearly",
106   "attribute1":"Online",
107   "attribute2":"Moving",
108   "attribute3":""
109 });
110
111 db.users.find();
```

Run

Result

```
{ "_id" : ObjectId("638be22c6cd264786516fab"), "user_id" : 1234, "user_name" : "AMAZON LLC", "user_type"
{ "_id" : ObjectId("638be22c6cd264786516fafc"), "user_id" : 2345, "user_name" : "LINCOLN COLN", "user_ty
{ "_id" : ObjectId("638be22c6cd264786516fefd"), "user_id" : 3456, "user_name" : "SKIPPER LLC", "user_type
{ "_id" : ObjectId("638be22c6cd264786516fafe"), "user_id" : 4567, "user_name" : "ZINC INC", "user_type"
{ "_id" : ObjectId("638be22c6cd264786516faff"), "user_id" : 5678, "user_name" : "ZOMBIE LLC", "user_type
{ "_id" : ObjectId("638be22c6cd264786516fb00"), "user_id" : 6789, "user_name" : "DOOR BUSS INC", "user_t
{ "_id" : ObjectId("638be22c6cd264786516fb01"), "user_id" : 7890, "user_name" : "AMAZON INC", "user_type
{ "_id" : ObjectId("638be22c6cd264786516fb02"), "user_id" : 8901, "user_name" : "CISCO LLC", "user_type"
{ "_id" : ObjectId("638be22c6cd264786516fb03"), "user_id" : 9012, "user_name" : "BEXLEY APARTMENTS", "us
{ "_id" : ObjectId("638be22c6cd264786516fb04"), "user_id" : 11234, "user_name" : "UHAUL LLC", "user_type
```

[-- Back to Playground overview](#)

Reset

[Click here to reset the database to its initial state](#) (all your changes will be lost).

Tips

Enter MongoDB Javascript commands in the text area. Pressing "Run" will present the result of the MongoDB shell output. Try

`db.getCollectionNames();` to see defined collections and
`db.COLLECTIONNAME.find();` to retrieve a list of documents inside the given collection. See the [MongoDB reference](#) for useful commands.

Customer Table:

```
db.customers.insert({
  "customer_id":1234,
  "user_id":4567,
  "customer_name":"AMAZON LLC",
  "gender": "",
  "dateofbirth": "April 15, 2000",
  "address": "9123 Amazon Way",
  "city": "San Jose",
  "state": "California",
  "zipcode": 95210,
  "phonenumber": "408-343-3545",
  "email":"amazonllc@amazon.com"
});
```

```
db.customers.insert({
  "customer_id":2345,
  "user_id":1234,
  "customer_name":"UHAUL LLC",
  "gender": "",
  "dateofbirth": "May 10, 1980",
  "address": "2344 Uhaul Way",
  "city": "San Jose",
  "state": "California",
  "zipcode": 95210,
  "phonenumber": "408-222-4535",
  "email":"support@uhaul.com"
});
```

```
db.customers.insert({
  "customer_id":3435,
  "user_id":7890,
  "customer_name":"JAMES BLACK",
  "gender": "Male",
  "dateofbirth": "January 15, 2002",
  "address": "2212 Tremont Way",
  "city": "Boston",
  "state": "Massachuseets",
  "zipcode": 03323,
  "phonenumber": "617-223-3545",
  "email":"jamesblack@gmail.com"
});
```

```
db.customers.insert({
```

```

    "customer_id":12344,
    "user_id":4342532,
    "customer_name":"CISCO INC",
    "gender": "",
    "dateofbirth": "June 01, 1900",
    "address": "9123 Cisco Way",
    "city": "San Jose",
    "state": "California",
    "zipcode": 95210,
    "phonenumber": "408-343-3545",
    "email":"user@cisco.com"
});

db.customers.insert({
    "customer_id":123674,
    "user_id":4567345,
    "customer_name":"CISCO SYSTEMS",
    "gender": "",
    "dateofbirth": "April 15, 2000",
    "address": "9123 Cisco Street",
    "city": "San Jose",
    "state": "California",
    "zipcode": 95233,
    "phonenumber": "408-111-3545",
    "email":"support@cisco.com"
});

db.customers.insert({
    "customer_id":87897,
    "user_id":4564,
    "customer_name":"HOBBY LOBBY",
    "gender": "",
    "dateofbirth": "April 15, 1920",
    "address": "9494 Hobby Way",
    "city": "Denver",
    "state": "Colorado",
    "zipcode": 95191,
    "phonenumber": "720-343-3545",
    "email":"hobby@gmail.com"
});

db.customers.insert({
    "customer_id":98997,
    "user_id":765765,
    "customer_name":"Sentra White",
    "gender": "Female",

```

```
"dateofbirth": "July 20, 2005",  
"address": "9123 Tremont Ave",  
"city": "Los Altos",  
"state": "California",  
"zipcode": 95110,  
"phonenumber": "408-959-3545",  
"email": "sentrawhite@gmail.com"  
});
```

```
db.customers.insert({  
  "customer_id": 34545,  
  "user_id": 342535,  
  "customer_name": "ZING LLC",  
  "gender": "",  
  "dateofbirth": "April 15, 2000",  
  "address": "9123 Zing Way",  
  "city": "Sacramento",  
  "state": "California",  
  "zipcode": 95230,  
  "phonenumber": "408-343-3545",  
  "email": "zingzing@gmail.com"  
});
```

```
db.customers.insert({  
  "customer_id": 7355,  
  "user_id": 2343,  
  "customer_name": "PINK INC",  
  "gender": "",  
  "dateofbirth": "April 02, 2002",  
  "address": "3455 Pink Way",  
  "city": "San Francisco",  
  "state": "California",  
  "zipcode": 92123,  
  "phonenumber": "972-343-3545",  
  "email": "support@pink.com"  
});
```

```
db.customers.insert({  
  "customer_id": 1234355,  
  "user_id": 4567345,  
  "customer_name": "MIKE BOSE",  
  "gender": "Male",  
  "dateofbirth": "August 15, 1934",  
  "address": "9123 Amazon Place",  
  "city": "Colorado Springs",  
  "state": "Colorado",  
});
```

```

"zipcode": 90414,
"phonenumber": "720-343-3545",
"email": "mikebose@bose.com"
});

```

```
db.customers.find();
```

Using: wine (mongo)

Write your statement below and press "Run" to see the result.

```

120      city: "San Francisco",
121      state: "California",
122      zipcode: 92123,
123      phonenumber: "972-343-3545",
124      email: "support@pink.com"
125    });
126
127 - db.customers.insert({
128   "customer_id": 1234355,
129   "user_id": 4567345,
130   "customer_name": "MIKE BOSE",
131   "gender": "Male",
132   "dateofbirth": "August 15, 1934",
133   "address": "9123 Amazon Place",
134   "city": "Colorado Springs",
135   "state": "Colorado",
136   "zipcode": 90414,
137   "phonenumber": "720-343-3545",
138   "email": "mikebose@bose.com"
139 });
140
141 db.customers.find();

```

Run

Result

```

{ "_id" : ObjectId("638beeda9eb767da7140ccad"), "customer_id" : 1234, "user_id" : 4567, "customer_name"
{ "_id" : ObjectId("638beeda9eb767da7140cace"), "customer_id" : 2345, "user_id" : 1234, "customer_name"
{ "_id" : ObjectId("638beeda9eb767da7140cacf"), "customer_id" : 3435, "user_id" : 7890, "customer_name"
{ "_id" : ObjectId("638beeda9eb767da7140cad0"), "customer_id" : 12344, "user_id" : 4342532, "customer_nam
{ "_id" : ObjectId("638beeda9eb767da7140cad1"), "customer_id" : 123674, "user_id" : 4567345, "customer_n
{ "_id" : ObjectId("638beeda9eb767da7140cad2"), "customer_id" : 87897, "user_id" : 4564, "customer_name"
{ "_id" : ObjectId("638beeda9eb767da7140cad3"), "customer_id" : 98997, "user_id" : 765765, "customer_nam
{ "_id" : ObjectId("638beeda9eb767da7140cad4"), "customer_id" : 34545, "user_id" : 342535, "customer_nam
{ "_id" : ObjectId("638beeda9eb767da7140cad5"), "customer_id" : 7355, "user_id" : 2343, "customer_name"

```

[Back to Playground overview](#)

Reset

[Click here to reset the database to its initial state](#) (all your changes will be lost).

Tips

Enter MongoDB Javascript commands in the text area. Pressing "Run" will present the result of the MongoDB shell output. Try

- `db.getCollectionNames();` to see defined collections and
- `db.COLLECTIONNAME.find();` to retrieve a list of documents inside the given collection. See the [MongoDB reference](#) for useful commands.

Business Table:

```

db.business.insert({
  "business_id": 1234,
  "user_id": 4567,
  "business_name": "AMAZON LLC",
  "typeofbusiness": "Business",
  "address": "9123 Amazon Way",
  "city": "San Jose",
  "state": "California",
  "zipcode": 95210,
  "phonenumber": "408-343-3545",
  "email": "amazonllc@amazon.com"
});

```

```
db.business.insert({
```



```

    "business_id":2345,
    "user_id":1234,
    "business_name":"UHAUL LLC",
    "typeofbusiness": "Business",
    "address": "2344 Uhaul Way",
    "city": "San Jose",
    "state": "California",
    "zipcode": 95210,
    "phonenumber": "408-222-4535",
    "email":"support@uhaul.com"
  });

db.business.insert({
  "business_id":3435,
  "user_id":7890,
  "business_name":"JAMES BLACK",
  "typeofbusiness": "Individual",
  "address": "2212 Tremont Way",
  "city": "Boston",
  "state": "Massachuseets",
  "zipcode": 03323,
  "phonenumber": "617-223-3545",
  "email":"jamesblack@gmail.com"
});

db.business.insert({
  "business_id":12344,
  "user_id":4342532,
  "business_name":"CISCO INC",
  "typeofbusiness": "Business",
  "address": "9123 Cisco Way",
  "city": "San Jose",
  "state": "California",
  "zipcode": 95210,
  "phonenumber": "408-343-3545",
  "email":"user@cisco.com"
});

db.business.insert({
  "business_id":123674,
  "user_id":4567345,
  "business_name":"CISCO SYSTEMS",
  "typeofbusiness": "Business",
  "address": "9123 Cisco Street",
  "city": "San Jose",
  "state": "California",

```

```
"zipcode": 95233,  
"phonenumber": "408-111-3545",  
"email": "support@cisco.com"  
});
```

```
db.business.insert({  
  "business_id": 87897,  
  "user_id": 4564,  
  "business_name": "HOBBY LOBBY",  
  "typeofbusiness": "Business",  
  "address": "9494 Hobby Way",  
  "city": "Denver",  
  "state": "Colorado",  
  "zipcode": 95191,  
  "phonenumber": "720-343-3545",  
  "email": "hobby@gmail.com"  
});
```

```
db.business.insert({  
  "business_id": 98997,  
  "user_id": 765765,  
  "business_name": "Sentra White",  
  "typeofbusiness": "Individual",  
  "address": "9123 Tremont Ave",  
  "city": "Los Altos",  
  "state": "California",  
  "zipcode": 95110,  
  "phonenumber": "408-959-3545",  
  "email": "sentrawhite@gmail.com"  
});
```

```
db.business.insert({  
  "business_id": 34545,  
  "user_id": 342535,  
  "business_name": "ZING LLC",  
  "typeofbusiness": "Business",  
  "address": "9123 Zing Way",  
  "city": "Sacramento",  
  "state": "California",  
  "zipcode": 95230,  
  "phonenumber": "408-343-3545",  
  "email": "zingzing@gmail.com"  
});
```

```
db.business.insert({  
  "business_id": 1234355,
```

```

"user_id":4567345,
"business_name":"MIKE BOSE",
  "typeofbusiness": "Individual",
"address": "9123 Amazon Place",
"city": "Colorado Springs",
"state": "Colorado",
"zipcode": 90414,
"phonenumber": "720-343-3545",
"email":"mikebose@bose.com"
});

```

```
db.business.find();
```

Using: wine (mongo)

Write your statement below and press "Run" to see the result.

```

1 - db.business.insert({
2   "business_id":1234,
3   "user_id":4567,
4   "business_name":"AMAZON LLC",
5   "typeofbusiness": "Business",
6   "address": "9123 Amazon Way",
7   "city": "San Jose",
8   "state": "California",
9   "zipcode": 95210,
10  "phonenumber": "408-343-3545",
11  "email":"amazonllc@amazon.com"
12 });
13
14 - db.business.insert({
15   "business_id":2345,
16   "user_id":1234,
17   "business_name":"UHAUL LLC",
18   "typeofbusiness": "Business",
19   "address": "2344 Uhaul Way",
20   "city": "San Jose",
21   "state": "California",
22   "zipcode": 95210
23 });

```

Run

Result

```

{ "_id" : ObjectId("638bf44632ee7761291dd4b5"), "business_id" : 1234, "user_id" : 4567, "business_name"
{ "_id" : ObjectId("638bf44632ee7761291dd4b6"), "business_id" : 2345, "user_id" : 1234, "business_name"
{ "_id" : ObjectId("638bf44632ee7761291dd4b7"), "business_id" : 3435, "user_id" : 7890, "business_name"
{ "_id" : ObjectId("638bf44632ee7761291dd4b8"), "business_id" : 12344, "user_id" : 4342532, "business_nam
{ "_id" : ObjectId("638bf44632ee7761291dd4b9"), "business_id" : 123674, "user_id" : 4567345, "business_n
{ "_id" : ObjectId("638bf44632ee7761291dd4ba"), "business_id" : 87897, "user_id" : 4564, "business_name"
{ "_id" : ObjectId("638bf44632ee7761291dd4bb"), "business_id" : 98997, "user_id" : 765765, "business_nam
{ "_id" : ObjectId("638bf44632ee7761291dd4bc"), "business_id" : 34545, "user_id" : 342535, "business_nam
{ "_id" : ObjectId("638bf44632ee7761291dd4bd"), "business_id" : 1234355, "user_id" : 4567345, "business_nam

```

[← Back to Playground overview](#)

Reset

[Click here to reset the database to its initial state](#) (all your changes will be lost).

Tips

Enter MongoDB Javascript commands in the text area. Pressing "Run" will present the result of the MongoDB shell output. Try `db.getCollectionNames();` to see defined collections and `db.COLLECTIONNAME.find();` to retrieve a list of documents inside the given collection. See the [MongoDB reference](#) for useful commands.

Product Table:

```
db.product.insert({
  "product_id":1234,
  "product_name":"TELEVISION",
  "product_color": "Black",
  "product_size": "Small",
  "available_quantity": 100,
  "product_price": 2000,
  "date_of_manufacture": "January 10, 2020",
  "date_of_expiry": "January 09, 2030"
});
```

```
db.product.insert({
  "product_id":435345,
  "product_name":"HAMPER",
  "product_color": "Black",
  "product_size": "LARGE",
  "available_quantity": 2100,
  "product_price": 10,
  "date_of_manufacture": "March 10, 2020",
  "date_of_expiry": "March 09, 2030"
});
```

```
db.product.insert({
  "product_id":878686,
  "product_name":"SUITCASE",
  "product_color": "PINK",
  "product_size": "Small",
  "available_quantity": 1300,
  "product_price": 150,
  "date_of_manufacture": "May 10, 2020",
  "date_of_expiry": "May 01, 2030"
});
```

```
db.product.insert({
  "product_id":97787,
  "product_name":"COMFORTER",
  "product_color": "Black",
  "product_size": "Medium",
  "available_quantity": 7100,
  "product_price": 50,
  "date_of_manufacture": "June 30, 2020",
  "date_of_expiry": "June 29, 2030"
});
```

```

db.product.insert({
  "product_id":5667,
  "product_name":"BOXES",
  "product_color": "GREY",
  "product_size": "Small",
  "available_quantity": 1000,
  "product_price": 10,
  "date_of_manufacture": "July 10, 2020",
  "date_of_expiry": "July 09, 2030"
});

```

```

db.product.find();

```

Using: wine (mongo)

Write your statement below and press "Run" to see the result.

```

1 db.product.insert({
2   "product_id":1234,
3   "product_name":"TELEVISION",
4   "product_color": "Black",
5   "product_size": "Small",
6   "available_quantity": 100,
7   "product_price": 2000,
8   "date_of_manufacture": "January 10, 2020",
9   "date_of_expiry": "January 09, 2030"
10 });
11
12 db.product.insert({
13   "product_id":435345,
14   "product_name":"HAMPER",
15   "product_color": "Black",
16   "product_size": "LARGE",
17   "available_quantity": 2100,
18   "product_price": 10,
19   "date_of_manufacture": "March 10, 2020",
20   "date_of_expiry": "March 09, 2030"
21 });
22

```

Run

Result

```

{ "_id" : ObjectId("638bf8e2a90f9418cc32d1e4"), "product_id" : 1234, "product_name" : "TELEVISION", "pro
{ "_id" : ObjectId("638bf8e2a90f9418cc32d1e5"), "product_id" : 435345, "product_name" : "HAMPER", "produ
{ "_id" : ObjectId("638bf8e2a90f9418cc32d1e6"), "product_id" : 878686, "product_name" : "SUITCASE", "pro
{ "_id" : ObjectId("638bf8e2a90f9418cc32d1e7"), "product_id" : 97787, "product_name" : "COMFORTER", "pro
{ "_id" : ObjectId("638bf8e2a90f9418cc32d1e8"), "product_id" : 5667, "product_name" : "BOXES", "product_

```

[← Back to Playground overview](#)

Reset

[Click here to reset the database to its initial state](#) (all your changes will be lost).

Tips

Enter MongoDB Javascript commands in the text area. Pressing "Run" will present the result of the MongoDB shell output. Try `db.getCollectionNames();` to see defined collections and `db.COLLECTIONNAME.find();` to retrieve a list of documents inside the given collection. See the [MongoDB reference](#) for useful commands.

Purchase Order Table:

```

db.purchase_order.insert({
  "purchase_id":1234,
  "purchase_method":"CARD",
  "po_date": "January 09, 2000",
  "product_id": 34535,
  "supplier_id": 34526
});

```

```
db.purchase_order.insert({
  "purchase_id":345466,
  "purchase_method":"CARD",
  "po_date": "May 09, 1990",
  "product_id": 867876,
  "supplier_id": 76765
});
```

```
db.purchase_order.insert({
  "purchase_id":987987,
  "purchase_method":"CASH",
  "po_date": "June 10, 2000",
  "product_id": 34535,
  "supplier_id": 100
});
```

```
db.purchase_order.insert({
  "purchase_id":9879,
  "purchase_method":"CARD",
  "po_date": "JULY 15, 2020",
  "product_id": 987987,
  "supplier_id": 6567
});
```

```
db.purchase_order.insert({
  "purchase_id":6464,
  "purchase_method":"CARD",
  "po_date": "August 09, 2007",
  "product_id": 987879,
  "supplier_id": 675
});
```

```
db.purchase_order.insert({
  "purchase_id":8786,
  "purchase_method":"CASH",
  "po_date": "September 09, 2009",
  "product_id": 7657,
  "supplier_id": 54354
});
```

```
db.purchase_order.insert({
  "purchase_id":9878,
  "purchase_method":"CARD",
  "po_date": "August 09, 2020",
  "product_id": 98798,
```

```

    "supplier_id": 6756
  });

db.purchase_order.insert({
  "purchase_id": 787687,
  "purchase_method": "CARD",
  "po_date": "January 20, 2001",
  "product_id": 767,
  "supplier_id": 564564
});

```

```
db.purchase_order.find();
```

Using: wine (mongo)

Write your statement below and press "Run" to see the result.

```

1 db.purchase_order.insert({
2   "purchase_id": 1234,
3   "purchase_method": "CARD",
4   "po_date": "January 09, 2000",
5   "product_id": 34535,
6   "supplier_id": 34526
7 });
8
9 db.purchase_order.insert({
10  "purchase_id": 345466,
11  "purchase_method": "CARD",
12  "po_date": "May 09, 1990",
13  "product_id": 867876,
14  "supplier_id": 76765
15 });
16
17 db.purchase_order.insert({
18  "purchase_id": 987987,
19  "purchase_method": "CASH",
20  "po_date": "June 10, 2000",
21  "product_id": 34535,
22  "supplier_id": 100
23 });

```

Run

Result

```

{ "_id" : ObjectId("638bfb4903c1c6aea8157e33"), "purchase_id" : 1234, "purchase_method" : "CARD", "po_da
{ "_id" : ObjectId("638bfb4903c1c6aea8157e34"), "purchase_id" : 345466, "purchase_method" : "CARD", "po_i
{ "_id" : ObjectId("638bfb4903c1c6aea8157e35"), "purchase_id" : 987987, "purchase_method" : "CASH", "po_i
{ "_id" : ObjectId("638bfb4903c1c6aea8157e36"), "purchase_id" : 9879, "purchase_method" : "CARD", "po_da
{ "_id" : ObjectId("638bfb4903c1c6aea8157e37"), "purchase_id" : 6464, "purchase_method" : "CARD", "po_da
{ "_id" : ObjectId("638bfb4903c1c6aea8157e38"), "purchase_id" : 8786, "purchase_method" : "CASH", "po_da
{ "_id" : ObjectId("638bfb4903c1c6aea8157e39"), "purchase_id" : 9878, "purchase_method" : "CARD", "po_da
{ "_id" : ObjectId("638bfb4903c1c6aea8157e3a"), "purchase_id" : 787687, "purchase_method" : "CARD", "po_i

```

[Back to Playground overview](#)

Reset

[Click here to reset the database to its initial state](#) (all your changes will be lost).

Tips

Enter MongoDB Javascript commands in the text area. Pressing "Run" will present the result of the MongoDB shell output. Try

- `db.getCollectionNames();` to see defined collections and
- `db.COLLECTIONNAME.find();` to retrieve a list of documents inside the given collection. See the [MongoDB reference](#) for useful commands.

PO Supp Table:

```

db.po_supp.insert({
  "po_supp": "2342-2344",
  "supp_po": "2344-2342"
});

```

```

db.po_supp.insert({
  "po_supp": "1234-2342",
  "supp_po": "2342-1234"
});

```

```
});
```

```
db.po_supp.insert({  
  "po_supp": "1234-6543",  
  "supp_po": "6543-1234"  
});
```

```
db.po_supp.insert({  
  "po_supp": "6543-1234",  
  "supp_po": "1234-6543"  
});
```

```
db.po_supp.insert({  
  "po_supp": "7789-6677",  
  "supp_po": "6677-7789"  
});
```

```
db.po_supp.insert({  
  "po_supp": "7654-2344",  
  "supp_po": "2344-7654"  
});
```

```
db.po_supp.insert({  
  "po_supp": "9988-2344",  
  "supp_po": "2344-9988"  
});
```

```
db.po_supp.find();
```


Using: wine (mongo)

Write your statement below and press "Run" to see the result.

```
1 db.po_supp.insert({
2   "po_supp": "2342-2344",
3   "supp_po": "2344-2342"
4 });
5
6 db.po_supp.insert({
7   "po_supp": "1234-2342",
8   "supp_po": "2342-1234"
9 });
10
11 db.po_supp.insert({
12   "po_supp": "1234-6543",
13   "supp_po": "6543-1234"
14 });
15
16 db.po_supp.insert({
17   "po_supp": "6543-1234",
18   "supp_po": "1234-6543"
19 });
20
21 db.po_supp.insert({
22   "po_supp": "7789-6677"
23 });
```

Run

Result

```
{ "_id" : ObjectId("638bff07671c0324ebffc460"), "po_supp" : "2342-2344", "supp_po" : "2344-2342" }
{ "_id" : ObjectId("638bff07671c0324ebffc461"), "po_supp" : "\t1234-2342", "supp_po" : "2342-1234" }
{ "_id" : ObjectId("638bff07671c0324ebffc462"), "po_supp" : "1234-6543", "supp_po" : "6543-1234" }
{ "_id" : ObjectId("638bff07671c0324ebffc463"), "po_supp" : "6543-1234", "supp_po" : "1234-6543" }
{ "_id" : ObjectId("638bff07671c0324ebffc464"), "po_supp" : "7789-6677", "supp_po" : "6677-7789" }
{ "_id" : ObjectId("638bff07671c0324ebffc465"), "po_supp" : "7654-2344", "supp_po" : "2344-7654" }
{ "_id" : ObjectId("638bff07671c0324ebffc466"), "po_supp" : "9988-2344", "supp_po" : "2344-9988" }
```

[← Back to Playground overview](#)

Reset

[Click here to reset the database to its initial state](#) (all your changes will be lost).

Tips

Enter MongoDB Javascript commands in the text area. Pressing "Run" will present the result of the MongoDB shell output. Try

`db.getCollectionNames();` to see defined collections and

`db.COLLECTIONNAME.find();` to retrieve a list of documents inside the given collection. See the [MongoDB reference](#) for useful commands.

Supplier Table:

```
db.Supplier.insert({
  "supplier_id":1234,
  "supplier_name":"AMAZON LLC",
  "supplier_address": "9123 Amazon Way",
  "supplier_city": "San Jose",
  "state": "California",
  "zipcode": 95210,
  "supplier_status": "Active"
});
```

```
db.supplier.insert({
  "supplier_id":2345,
  "supplier_name":"UHAUL LLC",
  "supplier_address": "2344 Uhaul Way",
  "supplier_city": "San Jose",
  "state": "California",
  "zipcode": 95210,
  "supplier_status": "Active"
});
```

```
db.supplier.insert({
  "supplier_id":3435,
```

```
    "supplier_name": "JAMES BLACK",
    "supplier_address": "2212 Tremont Way",
    "supplier_city": "Boston",
    "state": "Massachuseets",
    "zipcode": 03323,
    "suppier_status": "Active"
  });
```

```
db.supplier.insert({
  "supplier_id": 12344,
  "supplier_name": "CISCO INC",
  "supplier_address": "9123 Cisco Way",
  "supplier_city": "San Jose",
  "state": "California",
  "zipcode": 95210,
  "suppier_status": "Active"
});
```

```
db.supplier.insert({
  "supplier_id": 123674,
  "supplier_name": "CISCO SYSTEMS",
  "supplier_address": "9123 Cisco Street",
  "supplier_city": "San Jose",
  "state": "California",
  "zipcode": 95233,
  "suppier_status": "Active"
});
```

```
db.supplier.insert({
  "supplier_id": 87897,
  "supplier_name": "HOBBY LOBBY",
  "supplier_address": "9494 Hobby Way",
  "supplier_city": "Denver",
  "state": "Colorado",
  "zipcode": 95191,
  "suppier_status": "Active"
});
```

```
db.supplier.insert({
  "supplier_id": 98997,
  "supplier_name": "Sentra White",
  "supplier_address": "9123 Tremont Ave",
  "supplier_city": "Los Altos",
  "state": "California",
  "zipcode": 95110,
  "suppier_status": "Active"
});
```

```
});
```

```
db.supplier.insert({
  "supplier_id":34545,
  "supplier_name":"ZING LLC",
  "supplier_address": "9123 Zing Way",
  "supplier_city": "Sacramento",
  "state": "California",
  "zipcode": 95230,
  "supplier_status": "Active"
});
```

```
db.supplier.insert({
  "supplier_id":1234355,
  "supplier_name":"MIKE BOSE",
  "supplier_address": "9123 Amazon Place",
  "supplier_city": "Colorado Springs",
  "state": "Colorado",
  "zipcode": 90414,
  "supplier_status": "Active"
});
```

```
db.supplier.find();
```

Using: wine (mongo)

Write your statement below and press "Run" to see the result.

```
1 db.supplier.insert({
2   "supplier_id":1234,
3   "supplier_name":"AMAZON LLC",
4   "supplier_address": "9123 Amazon Way",
5   "supplier_city": "San Jose",
6   "state": "California",
7   "zipcode": 95210,
8   "supplier_status": "Active"
9 });
10
11 db.supplier.insert({
12   "supplier_id":2345,
13   "supplier_name":"UHAUL LLC",
14   "supplier_address": "2344 Uhaul Way",
15   "supplier_city": "San Jose",
16   "state": "California",
17   "zipcode": 95210,
18   "supplier_status": "Active"
19 });
20
21 db.supplier.insert({
22   "supplier_id":3435
```

Run

Result

```
{ "_id" : ObjectId("638c038d135f2c70239eae2"), "supplier_id" : 2345, "supplier_name" : "UHAUL LLC", "su
{ "_id" : ObjectId("638c038d135f2c70239eae3"), "supplier_id" : 3435, "supplier_name" : "JAMES BLACK", "
{ "_id" : ObjectId("638c038d135f2c70239eae4"), "supplier_id" : 12344, "supplier_name" : "CISCO INC", "s
{ "_id" : ObjectId("638c038d135f2c70239eae5"), "supplier_id" : 123674, "supplier_name" : "CISCO SYSTEMS
{ "_id" : ObjectId("638c038d135f2c70239eae6"), "supplier_id" : 87897, "supplier_name" : "HOBBY LOBBY",
{ "_id" : ObjectId("638c038d135f2c70239eae7"), "supplier_id" : 98997, "supplier_name" : "Sentra White",
{ "_id" : ObjectId("638c038d135f2c70239eae8"), "supplier_id" : 34545, "supplier_name" : "ZING LLC", "su
{ "_id" : ObjectId("638c038d135f2c70239eae9"), "supplier_id" : 1234355, "supplier_name" : "MIKE BOSE",
```

[← Back to Playground overview](#)

Reset

[Click here to reset the database to its initial state](#) (all your changes will be lost).

Tips

Enter MongoDB Javascript commands in the text area. Pressing "Run" will present the result of the MongoDB shell output. Try

`db.getCollectionNames();` to see defined collections and

`db.COLLECTIONNAME.find();` to retrieve a list of documents inside the given collection. See the [MongoDB reference](#) for useful commands.

Customer Order Table:

```
db.customer_order.insert({  
  "order_id":1234,  
  "order_date": "June 15, 2020",  
  "address": "9123 Amazon Way, San Jose, California, 95210",  
  "delivery_estimate": "June 30, 2020"  
});
```

```
db.customer_order.insert({  
  "order_id":2345,  
  "order_date": "August 20, 2020",  
  "address": "2344 Uhaul Way, San Jose, California, 95210",  
  "delivery_estimate": "August 30, 2020"  
});
```

```
db.customer_order.insert({  
  "order_id":3435,  
  "order_date": "July 20, 2020",  
  "address": "2212 Tremont Way, Boston, Massachuseets, 03323",  
  "delivery_estimate": "July 30, 2020"  
});
```

```
db.customer_order.insert({  
  "order_id":12344,  
  "order_date": "July 01, 2022",  
  "address": "9123 Cisco Way, San Jose, California, 95210",  
  "delivery_estimate": "July 05, 2022"  
});
```

```
db.customer_order.insert({  
  "order_id":123674,  
  "order_date": "August 01, 2000",  
  "address": "9123 Cisco Street, San Jose, California, 95233",  
  "delivery_estimate": "August 05, 2000"  
});
```

```
db.customer_order.insert({  
  "order_id":87897,  
  "order_date": "September 01, 2003",  
  "address": "9494 Hobby Way, Denver, Colorado, 95191",  
  "delivery_estimate": "September 05, 2003"  
});
```

```
db.customer_order.insert({  
  "order_id":98997,
```

```

    "order_date": "May 01, 2003",
    "address": "9123 Tremont Ave, Los Altos, California, 95110",
    "delivery_estimate": "May 05, 2003"
  });

```

```
db.customer_order.find();
```

Using: wine (mongo)

Write your statement below and press "Run" to see the result.

```

1 db.customer_order.insert({
2   "order_id":1234,
3   "order_date": "June 15, 2020",
4   "address": "9123 Amazon Way, San Jose, California, 95210",
5   "delivery_estimate": "June 30, 2020"
6 });
7
8 db.customer_order.insert({
9   "order_id":2345,
10  "order_date": "August 20, 2020",
11  "address": "2344 Uhaul Way, San Jose, California, 95210",
12  "delivery_estimate": "August 30, 2020"
13 });
14
15 db.customer_order.insert({
16   "order_id":3435,
17   "order_date": "July 20, 2020",
18   "address": "2212 Tremont Way, Boston, Massachusetts, 03323",
19   "delivery_estimate": "July 30, 2020"
20 });
21
22 db.customer_order.insert({

```

Run

Result

```

{ "_id" : ObjectId("638c08a6a6096850e032bd59"), "order_id" : 1234, "order_date" : "June 15, 2020", "addr
{ "_id" : ObjectId("638c08a6a6096850e032bd5a"), "order_id" : 2345, "order_date" : "August 20, 2020", "adi
{ "_id" : ObjectId("638c08a6a6096850e032bd5b"), "order_id" : 3435, "order_date" : "July 20, 2020", "addr
{ "_id" : ObjectId("638c08a6a6096850e032bd5c"), "order_id" : 12344, "order_date" : "July 01, 2022", "addi
{ "_id" : ObjectId("638c08a6a6096850e032bd5d"), "order_id" : 123674, "order_date" : "August 01, 2000", ".
{ "_id" : ObjectId("638c08a6a6096850e032bd5e"), "order_id" : 87897, "order_date" : "September 01, 2003",
{ "_id" : ObjectId("638c08a6a6096850e032bd5f"), "order_id" : 98997, "order_date" : "May 01, 2003", "addr

```

[← Back to Playground overview](#)

Reset

[Click here to reset the database to its initial state](#) (all your changes will be lost).

Tips

Enter MongoDB Javascript commands in the text area. Pressing "Run" will present the result of the MongoDB shell output. Try `db.getCollectionNames();` to see defined collections and `db.COLLECTIONNAME.find();` to retrieve a list of documents inside the given collection. See the [MongoDB reference](#) for useful commands.

Warehouse Table:

```

db.warehouses.insert({
  "warehouse_id":1234,
  "warehouse_address": "9123 Amazon Way, San Jose, California, 95210",
  "phonenummer": "408-343-3545"
});

```

```

db.warehouses.insert({
  "warehouse_id":68876,
  "warehouse_address": "2344 Uhaul Way, San Jose, California, 95210",
  "phonenummer": "408-222-3545"
});

```

```

db.warehouses.insert({
  "warehouse_id":987897,
  "warehouse_address": "2212 Tremont Way, Boston, Massachusetts, 03323",

```

```

    "phonenumner": "617-223-3545"
  });

db.warehouses.insert({
  "warehouse_id":5465,
  "warehouse_address": "9123 Cisco Way, San Jose, California, 95210",
  "phonenumner": "408-343-3545"
});

db.warehouses.insert({
  "warehouse_id":324243,
  "warehouse_address": "9123 Cisco Street, San Jose, California, 95233",
  "phonenumner": "408-111-3545"
});

db.warehouses.insert({
  "warehouse_id":98987,
  "warehouse_address": "9494 Hobby Way, Denver, Colorado, 95191",
  "phonenumner": "720-111-8978"
});

db.warehouses.insert({
  "warehouse_id":6567,
  "warehouse_address": "9123 Tremont Ave, Los Altos, California, 95110",
  "phonenumner": "408-111-7676"
});

db.warehouses.find();

```

Using: wine (mongo)

Write your statement below and press "Run" to see the result.

```

1 db.warehouses.insert({
2   "warehouse_id":1234,
3   "warehouse_address": "9123 Amazon Way, San Jose, California, 95210",
4   "phonenumner": "408-343-3545"
5 });
6
7 db.warehouses.insert({
8   "warehouse_id":68876,
9   "warehouse_address": "2344 Uhaul Way, San Jose, California, 95210",
10  "phonenumner": "408-222-3545"
11 });
12
13 db.warehouses.insert({
14   "warehouse_id":987897,
15   "warehouse_address": "2212 Tremont Way, Boston, Massachusetts, 03123",
16   "phonenumner": "617-223-3545"
17 });
18
19 db.warehouses.insert({
20   "warehouse_id":5465,
21   "warehouse_address": "9123 Cisco Way, San Jose, California, 95210",
22   "phonenumner": "408-343-3545"
23 });

```

Run

Result

```

{ "_id" : ObjectId("638c0fae7a05b4ca09861bcd"), "warehouse_id" : 1234, "warehouse_address" : "9123 Amazon
{ "_id" : ObjectId("638c0fae7a05b4ca09861bce"), "warehouse_id" : 68876, "warehouse_address" : "2344 Uhaul
{ "_id" : ObjectId("638c0fae7a05b4ca09861bcf"), "warehouse_id" : 987897, "warehouse_address" : "2212 Trei
{ "_id" : ObjectId("638c0fae7a05b4ca09861bd0"), "warehouse_id" : 5465, "warehouse_address" : "9123 Cisco
{ "_id" : ObjectId("638c0fae7a05b4ca09861bd1"), "warehouse_id" : 324243, "warehouse_address" : "9123 Cis
{ "_id" : ObjectId("638c0fae7a05b4ca09861bd2"), "warehouse_id" : 98987, "warehouse_address" : "9494 Hobb
{ "_id" : ObjectId("638c0fae7a05b4ca09861bd3"), "warehouse_id" : 6567, "warehouse_address" : "9123 Tremo

```

[← Back to Playground overview](#)

Reset

[Click here to reset the database to its initial state](#) (all your changes will be lost).

Tips

Enter MongoDB Javascript commands in the text area. Pressing "Run" will present the result of the MongoDB shell output. Try `db.getCollectionNames();` to see defined collections and `db.COLLECTIONNAME.find();` to retrieve a list of documents inside the given collection. See the [MongoDB reference](#) for useful commands.

Logistics Table:

```
db.logistics.insert({  
  "logistics_id":1234,  
  "type_of_logistics": "USPS",  
  "tracking_number": 97867352431425678  
});
```

```
db.logistics.insert({  
  "logistics_id":1234,  
  "type_of_logistics": "UPS",  
  "tracking_number": "1ZW67352431425678"  
});
```

```
db.logistics.insert({  
  "logistics_id":1234,  
  "type_of_logistics": "FEDEX",  
  "tracking_number": "67564856444"  
});
```

```
db.logistics.insert({  
  "logistics_id":1234,  
  "type_of_logistics": "DHL",  
  "tracking_number": "675975757576597566"  
});
```

```
db.logistics.insert({  
  "logistics_id":1234,  
  "type_of_logistics": "USPS",  
  "tracking_number": "876795464865456845747"  
});
```

```
db.logistics.find();
```

Using: wine (mongo)

Write your statement below and press "Run" to see the result.

```
1 db.logistics.insert({
2   "logistics_id":1234,
3   "type_of_logistics": "USPS",
4   "tracking_number": 97867352431425678
5 });
6
7 db.logistics.insert({
8   "logistics_id":1234,
9   "type_of_logistics": "UPS",
10  "tracking_number": "1Zw67352431425678"
11 });
12
13 db.logistics.insert({
14   "logistics_id":1234,
15   "type_of_logistics": "FEDEX",
16   "tracking_number": "67564856444"
17 });
18
19 db.logistics.insert({
20   "logistics_id":1234,
21   "type_of_logistics": "DHL",
22   "tracking_number": "675075757576507566"
```

Run

Result

```
{ "_id" : ObjectId("638c11fceb92067c363f60"), "logistics_id" : 1234, "type_of_logistics" : "USPS", "tr
{ "_id" : ObjectId("638c11fceb92067c363f61"), "logistics_id" : 1234, "type_of_logistics" : "UPS", "tra
{ "_id" : ObjectId("638c11fceb92067c363f62"), "logistics_id" : 1234, "type_of_logistics" : "FEDEX", "t
{ "_id" : ObjectId("638c11fceb92067c363f63"), "logistics_id" : 1234, "type_of_logistics" : "DHL", "tra
{ "_id" : ObjectId("638c11fceb92067c363f64"), "logistics_id" : 1234, "type_of_logistics" : "USPS", "tr
```

[← Back to Playground overview](#)

Reset

[Click here to reset the database to its initial state](#) (all your changes will be lost).

Tips

Enter MongoDB Javascript commands in the text area. Pressing "Run" will present the result of the MongoDB shell output. Try `db.getCollectionNames();` to see defined collections and `db.COLLECTIONNAME.find();` to retrieve a list of documents inside the given collection. See the [MongoDB reference](#) for useful commands.

Local Logistics Table:

```
db.local_logistics.insert({
  "logistics_local_id":9689,
  "logistics_id": 8767956,
  "Shipper_address": "9123 Amazon Way, San Jose, California, 95210" ,
  "shipper_code": "USPS"
});
```

```
db.local_logistics.insert({
  "logistics_local_id":8768,
  "logistics_id": 989790,
  "Shipper_address": "2344 Uhaul Way, San Jose, California, 95210" ,
  "shipper_code": "FEDEX"
});
```

```
db.local_logistics.insert({
  "logistics_local_id":4535,
  "logistics_id": 98678596,
  "Shipper_address": "2212 Tremont Way, Boston, Massachuseets, 03323" ,
  "shipper_code": "UPS"
});
```

```
db.local_logistics.insert({
  "logistics_local_id":87678,
```



```

    "logistics_id": 654575475,
    "Shipper_address": "9494 Hobby Way, Denver, Colorado, 95191" ,
    "shipper_code": "UPS"
  });

```

```
db.local_logistics.find();
```

Using: wine (mongo)

Write your statement below and press "Run" to see the result.

```

1 db.local_logistics.insert({
2   "logistics_local_id": 9689,
3   "logistics_id": 8767956,
4   "Shipper_address": "9123 Amazon Way, San Jose, California, 95210" ,
5   "shipper_code": "USPS"
6 });
7
8 db.local_logistics.insert({
9   "logistics_local_id": 8768,
10  "logistics_id": 989790,
11  "Shipper_address": "2344 Uhaul Way, San Jose, California, 95210" ,
12  "shipper_code": "FEDEX"
13 });
14
15 db.local_logistics.insert({
16   "logistics_local_id": 4535,
17   "logistics_id": 98678596,
18   "Shipper_address": "2212 Tremont Way, Boston, Massachusetts, 03323" ,
19   "shipper_code": "UPS"
20 });
21
22 db.local_logistics.insert({

```

[Run](#)

Result

```

{ "_id" : ObjectId("638c1340b34ee475da5f0cee"), "logistics_local_id" : 9689, "logistics_id" : 8767956, "Shipper_address": "9123 Amazon Way, San Jose, California, 95210" , "shipper_code": "USPS" }
{ "_id" : ObjectId("638c1340b34ee475da5f0cef"), "logistics_local_id" : 8768, "logistics_id" : 989790, "Shipper_address": "2344 Uhaul Way, San Jose, California, 95210" , "shipper_code": "FEDEX" }
{ "_id" : ObjectId("638c1340b34ee475da5f0cf0"), "logistics_local_id" : 4535, "logistics_id" : 98678596, "Shipper_address": "2212 Tremont Way, Boston, Massachusetts, 03323" , "shipper_code": "UPS" }
{ "_id" : ObjectId("638c1340b34ee475da5f0cf1"), "logistics_local_id" : 87678, "logistics_id" : 654575475, "Shipper_address": "9494 Hobby Way, Denver, Colorado, 95191" , "shipper_code": "UPS" }

```

[← Back to Playground overview](#)

Reset

[Click here to reset the database to its initial state](#) (all your changes will be lost).

Tips

Enter MongoDB Javascript commands in the text area. Pressing "Run" will present the result of the MongoDB shell output. Try `db.getCollectionNames();` to see defined collections and `db.COLLECTIONNAME.find();` to retrieve a list of documents inside the given collection. See the [MongoDB reference](#) for useful commands.

International Table:

```

db.international.insert({
  "intl_logistics_id": 2134532,
  "intl_Shipper_address": "2-21 Delhi Way, New Delhi, India" ,
  "intl_shipper_code": "GARUDA VEGA"
});

```

```

db.international.insert({
  "intl_logistics_id": 54364334,
  "intl_Shipper_address": "3212 Hyderabad, Telangana, India" ,
  "intl_shipper_code": "FEDEX"
});

```

```

db.international.insert({
  "intl_logistics_id": 7658586584,
  "intl_Shipper_address": "43-334-23 Ontario, Canada" ,
  "intl_shipper_code": "UPS"
});

```

```
});
```

```
db.international.insert({
  "intl_logistics_id": 876798675,
  "intl_Shipper_address": "1/2/3 Paris, France" ,
  "intl_shipper_code": "USPS"
});
```

```
db.international.find();
```

Using: wine (mongo)

Write your statement below and press "Run" to see the result.

```
1 db.international.insert({
2   "intl_logistics_id": 2134532,
3   "intl_Shipper_address": "2-21 Delhi Way, New Delhi, India" ,
4   "intl_shipper_code": "GARUDA VEGA"
5 });
6
7 db.international.insert({
8   "intl_logistics_id": 54364334,
9   "intl_Shipper_address": "3212 Hyderabad, Telangana, India" ,
10  "intl_shipper_code": "FEDEX"
11 });
12
13 db.international.insert({
14   "intl_logistics_id": 7658586584,
15   "intl_Shipper_address": "43-334-23 Ontario, Canada" ,
16   "intl_shipper_code": "UPS"
17 });
18
19 db.international.insert({
20   "intl_logistics_id": 876798675,
21   "intl_Shipper_address": "1/2/3 Paris, France" ,
22   "intl_shipper_code": "USPS"
23 });
```

[Run](#)

Result

```
{ "_id" : ObjectId("638c148fa2458d68e043f067"), "intl_logistics_id" : 2134532, "intl_Shipper_address" : "2-21 Delhi Way, New Delhi, India" , "intl_shipper_code" : "GARUDA VEGA" }
{ "_id" : ObjectId("638c148fa2458d68e043f068"), "intl_logistics_id" : 54364334, "intl_Shipper_address" : "3212 Hyderabad, Telangana, India" , "intl_shipper_code" : "FEDEX" }
{ "_id" : ObjectId("638c148fa2458d68e043f069"), "intl_logistics_id" : 7658586584, "intl_Shipper_address" : "43-334-23 Ontario, Canada" , "intl_shipper_code" : "UPS" }
{ "_id" : ObjectId("638c148fa2458d68e043f06a"), "intl_logistics_id" : 876798675, "intl_Shipper_address" : "1/2/3 Paris, France" , "intl_shipper_code" : "USPS" }
```

[← Back to Playground overview](#)

Reset

[Click here to reset the database to its initial state](#) (all your changes will be lost).

Tips

Enter MongoDB Javascript commands in the text area. Pressing "Run" will present the result of the MongoDB shell output. Try `db.getCollectionNames();` to see defined collections and `db.COLLECTIONNAME.find();` to retrieve a list of documents inside the given collection. See the [MongoDB reference](#) for useful commands.

2. Update Statement in NoSQL:

Before Update:

[Run](#)

Result

```
{ "_id" : ObjectId("638c038d135f2c70239eaeef2"), "supplier_id" : 2345, "supplier_name" : "UHAUL LLC", "supplier_address" : "1234 Main St, New York, NY 10001" }
{ "_id" : ObjectId("638c038d135f2c70239eaeef3"), "supplier_id" : 3435, "supplier_name" : "JAMES BLACK", "supplier_address" : "5678 Main St, New York, NY 10001" }
{ "_id" : ObjectId("638c038d135f2c70239eaeef4"), "supplier_id" : 12344, "supplier_name" : "CISCO INC", "supplier_address" : "9876 Main St, New York, NY 10001" }
{ "_id" : ObjectId("638c038d135f2c70239eaeef5"), "supplier_id" : 123674, "supplier_name" : "CISCO SYSTEMS", "supplier_address" : "5432 Main St, New York, NY 10001" }
{ "_id" : ObjectId("638c038d135f2c70239eaeef6"), "supplier_id" : 87897, "supplier_name" : "HOBBY LOBBY", "supplier_address" : "1098 Main St, New York, NY 10001" }
{ "_id" : ObjectId("638c038d135f2c70239eaeef7"), "supplier_id" : 98997, "supplier_name" : "Sentra White", "supplier_address" : "2109 Main St, New York, NY 10001" }
{ "_id" : ObjectId("638c038d135f2c70239eaeef8"), "supplier_id" : 34545, "supplier_name" : "ZING LLC", "supplier_address" : "3210 Main St, New York, NY 10001" }
{ "_id" : ObjectId("638c038d135f2c70239eaeef9"), "supplier_id" : 1234355, "supplier_name" : "MIKE BOSE", "supplier_address" : "4321 Main St, New York, NY 10001" }
```

the given collection. See the [MongoDB reference](#) for useful commands.

```
db.supplier.update(
  {"supplier_id":3435},
  {$set:
    {"supplier_name": "JAMES MATT WHITE"}}
);
```

```
db.supplier.find();
```

After Update:

Using: wine (mongo)

Write your statement below and press "Run" to see the result.

```
1 db.supplier.update(
2   {"supplier_id":3435},
3   {$set:
4     {"supplier_name": "JAMES MATT WHITE"}}
5   );
6
7 db.supplier.find();
```

[Run](#)

Result

```
_id : ObjectId("638c038d135f2c70239eae2"), "supplier_id" : 2345, "supplier_name" : "UHAUL LLC", "suppl
_id" : ObjectId("638c038d135f2c70239eae3"), "supplier_id" : 3435, "supplier_name" : "JAMES MATT WHITE",
_id" : ObjectId("638c038d135f2c70239eae4"), "supplier_id" : 12344, "supplier_name" : "CISCO INC", "supp
_id" : ObjectId("638c038d135f2c70239eae5"), "supplier_id" : 123674, "supplier_name" : "CISCO SYSTEMS",
_id" : ObjectId("638c038d135f2c70239eae6"), "supplier_id" : 87897, "supplier_name" : "HOBBY LOBBY", "su
_id" : ObjectId("638c038d135f2c70239eae7"), "supplier_id" : 98997, "supplier_name" : "Sentra White", "s
_id" : ObjectId("638c038d135f2c70239eae8"), "supplier_id" : 34545, "supplier_name" : "ZING LLC", "suppl
_id" : ObjectId("638c038d135f2c70239eae9"), "supplier_id" : 1234355, "supplier_name" : "MIKE BOSE", "su
```

[← Back to Playground overview](#)

Reset

[Click here to reset the database to its initial state](#) (all your changes will be lost).

Tips

Enter MongoDB Javascript commands in the text area. Pressing "Run" will present the result of the MongoDB shell output. Try

- `db.getCollectionNames();` to see defined collections and
- `db.COLLECTIONNAME.find();` to retrieve a list of documents inside the given collection. See the [MongoDB reference](#) for useful commands.

3. Example of a medium size query like using aggregation:

Code:

```
db.product.aggregate(
  [{ $group: { _id: "$product_size", total: { $sum: "$product_price" } } },
  { $sort: { total: 1 } }
]);
```

Using: wine (mongo)

Write your statement below and press "Run" to see the result.

```
1 db.product.aggregate(  
2   [{ $group: { _id: "$product_size", total: { $sum: "$product_price" } } },  
3   { $sort: { total: 1 } }  
4 ] );
```

Run

Result

```
{ "_id" : "LARGE", "total" : 20 }  
{ "_id" : "Medium", "total" : 100 }  
{ "_id" : "Small", "total" : 4320 }
```

[← Back to Playground overview](#)

Reset

[Click here to reset the database to its initial state](#) (all your changes will be lost).

Tips

Enter MongoDB Javascript commands in the text area. Pressing "Run" will present the result of the MongoDB shell output. Try `db.getCollectionNames();` to see defined collections and `db.COLLECTIONNAME.find();` to retrieve a list of documents inside the given collection. See the [MongoDB reference](#) for useful commands.