CSCI235 Database Systems

MongoDB Query Language

Dr Janusz R. Getta

School of Computing and Information Technology -
University of Wollongong

MongoDB Query language

Outline

MongoDB query language

A sample database

Simple queries

Queries with Boolean operations

Queries on nested documents

Queries on arrays

Projections

Queries about `NULL`s and missing keys

Iterations over a cursor

MongoDB query language

MongoDB query language is based on a concept of pattern matching

A query is expressed as a BSON pattern and all documents that match the pattern are included in an answer

A method `find()` can be used to match a pattern with the documents in a collection `orders`

```
db.orders.find({"_id":"ALFKI"})
```
find()

Matching of an empty pattern `{}` with a collection `orders` returns an entire collection

```
db.orders.find({})
```
find()

Finding the first 3 documents in a collection `orders`

```
db.orders.find({}).limit(3)
```
find()

Finding all documents in a collection `orders` and listing the results in a nice format

```
db.orders.find({}).pretty()
```
find()

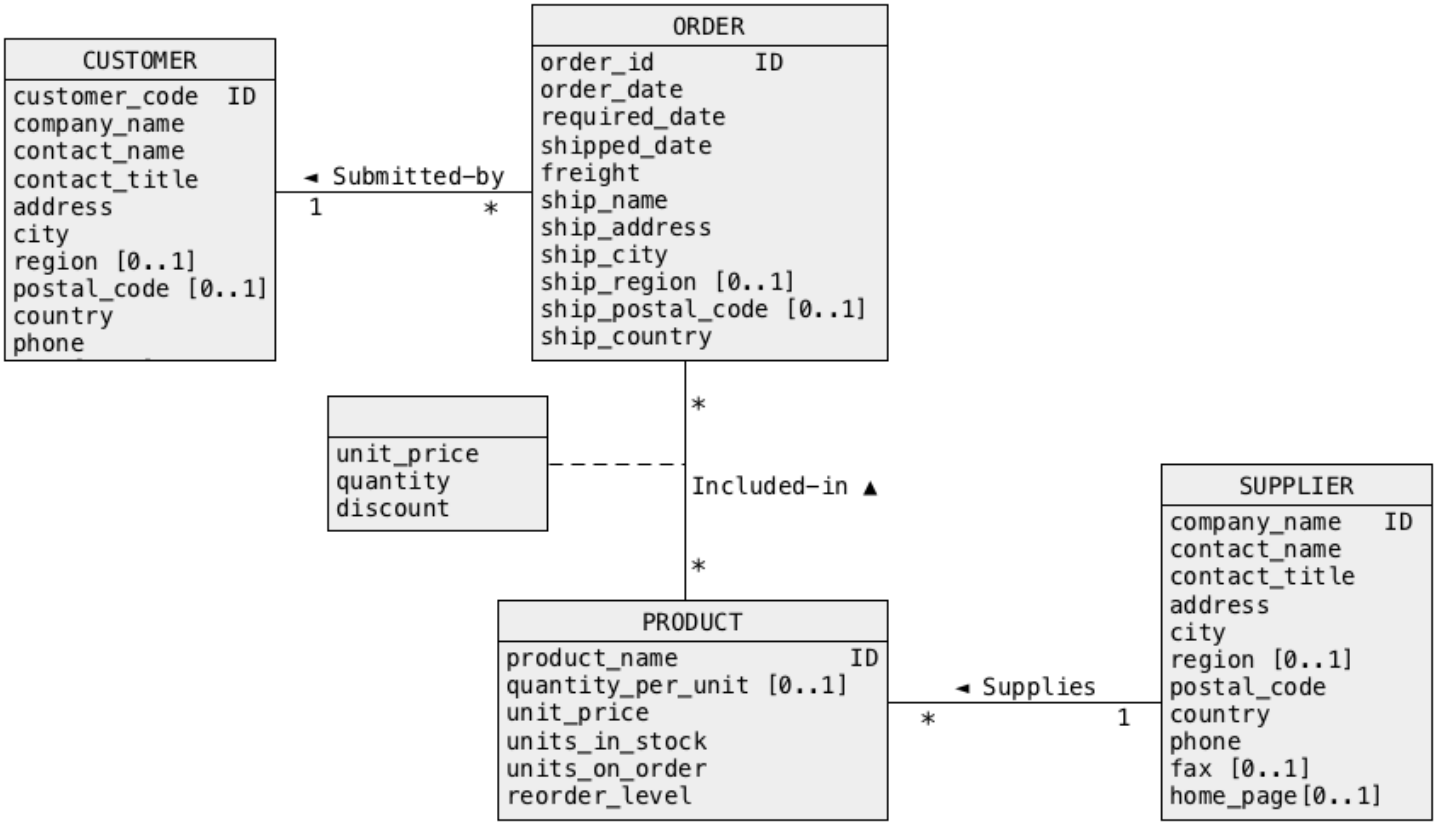
MongoDB Query language

Outline

MongoDB query language
A sample database
Simple queries
Queries with Boolean operations
Queries on nested documents
Queries on arrays
Projections
Queries about `NULL`s and missing keys
Iterations over a cursor

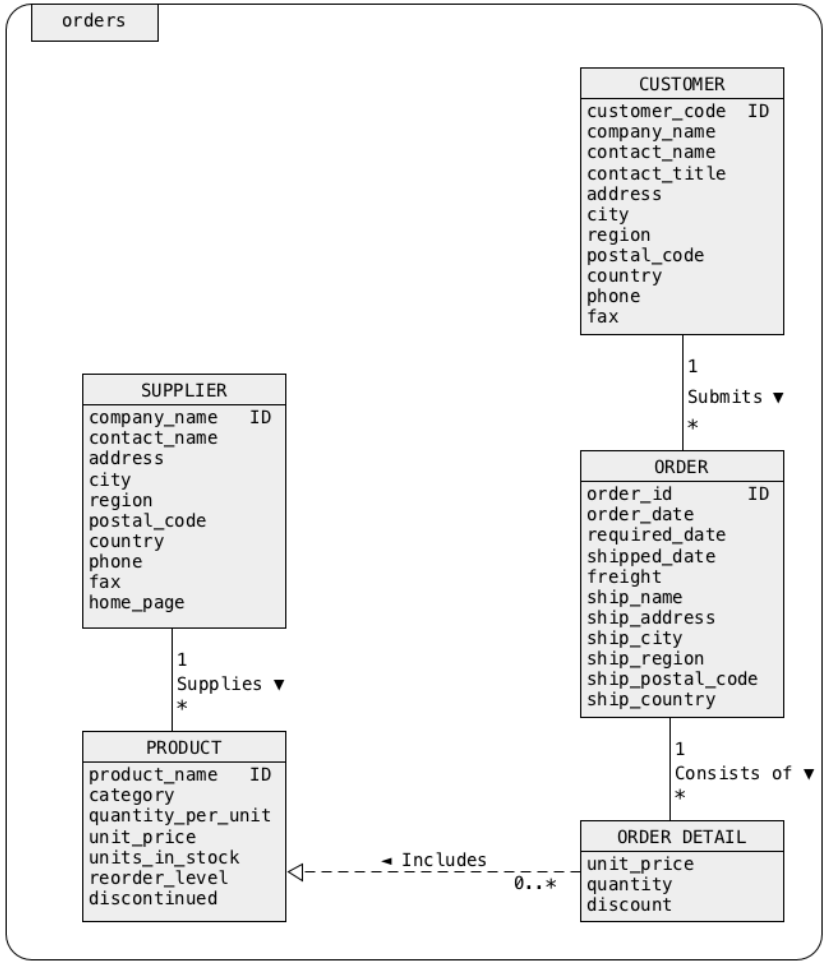
A sample database

A conceptual schema of a database with information about suppliers, products, customers, orders, and details of orders



A sample database

A sample collection `orders`



In HTML view press 'p' to see the lecture notes

TOP

A sample database

A sample document, that belongs to a class `CUSTOMER`

CUSTOMER

```
{
        "_id" : "ALFKI",
        "CUSTOMER" : {
                "customer code" : "ALFKI",
                "company name" : "Alfreds Futterkiste",
                "contact name" : "Maria Anders",
                "contact title" : "Sales Representative",
                "address" : "Obere Str. 57",
                "city" : "Berlin",
                "region" : null,
                "postal code" : "12209",
                "country" : "Germany",
                "phone" : "030-0074321",
                "fax" : "030-0076545",
                "submits" : [ ]
        }
}
```

A sample database

A sample nested document, that belongs to a class `CUSTOMER`

CUSTOMER

```
{"_id" : "FAMIA",
 "CUSTOMER" : {
              "customer code" : "FAMIA",
               ... ... ... ... ... ...
              "submits" : [
                          {
                           "ORDER" : {
                                     "order id" : 328,
                                      ... ... ... ... ...
                                     "consists of" : [
                                                     {
                                                      "ORDER DETAIL" : {
                                                                      "product name" : "Louisiana Fiery Hot Pepper Sauce",
                                                                       ... ... ... ... ...
                                                                      }
                                                     },
                                                     {
                                                      "ORDER DETAIL" : {
                                                                      "product name" : "Raclette Courdavault",
                                                                       ... ... ... ... ...
                                                                      }
                                                     }
                                                    ]
                                     }
                           }
                          ]
             } }
```

In HTML view press 'p' to see the lecture notes
TOP

A sample database

A sample nested document, that belongs to a class `SUPPLIER`

SUPPLIER

```
{
 "_id" : "Karkki Oy",
 "SUPPLIER" : {
              "company name" : "Karkki Oy",
              "contact name" : "Anne Heikkonen",
              "contact title" : "Product Manager",
              "address" : "Valtakatu 12",
              ...  ...  ...  ...  ...  ...
              "supplies" : [
                           {
                            "PRODUCT" : {
                                        "product name" : "Maxilaku",
                                        "category name" : "Confections",
                                         ...  ...  ...  ...  ...
                                       }
                           },
                           {
                            "PRODUCT" : {
                                        "product name" : "Valkoinen suklaa",
                                        "category name" : "Confections",
                                        ...  ...  ...  ...  ...
                                       }
                           }

                          ]
             }
}
```

In HTML view press 'p' to see the lecture notes

MongoDB Query language

Outline

MongoDB query language
A sample database
Simple queries
Queries with Boolean operations
Queries on nested documents
Queries on arrays
Projections
Queries about `NULL`s and missing keys
Iterations over a cursor

Simple queries

Find total number of documents in a collection

```
db.orders.count()                                                    count()
```

Find all information about all customers

```
db.orders.find({"CUSTOMER":{$exists:true}})                          Find entire class
```

Find all information about all suppliers

```
db.orders.find({"SUPPLIER":{$exists:true}})                          Find entire class
```

Find all information about the customers living in `Germany`

```
db.orders.find({"CUSTOMER.country":"Germany"})                       Access path
```

Find all information about the suppliers living in a city of `Oviedo`

```
db.orders.find({"SUPPLIER.city":"Oviedo"})                           Access path
```

Find all information about the suppliers who live in the `Netherlands` and have a contact title `Accounting Manager`

```
db.orders.find({"SUPPLIER.country":"Netherlands",                    And
                "SUPPLIER.contact title":"Accounting Manager"})
```

In HTML view press 'p' to see the lecture notes
TOP

MongoDB Query language

Outline

MongoDB query language
A sample database
Simple queries
Queries with Boolean operations
Queries on nested documents
Queries on arrays
Projections
Queries about `NULL`s and missing keys
Iterations over a cursor

Queries with Boolean operations

Comparison "key"="value"

```
{"key":"value"}
```
=

```
{"key":{$eq:"value"}}
```
=

Comparison "key" > "value"

```
{"key":{$gt:"value"}}
```
>

Disjunction ("key1"="value1") or ("key2"="value2")

```
{$or:[{"key1":"value1"},{"key2":"value2"}]}
```
$or

Conjunction ("key1"="value1") and ("key2"="value2")

```
{$and:[{"key1":"value1"},{"key2":"value2"}]}
```
$and

Boolean expression (("key1"="value1") or ("key2"="value2")) and ("key3"="value3")

Boolean expression

```
{$and:[{$or:[{"key1":"value1"},{"key2":"value2"}]},{"key3":"value3"}]}
```

In HTML view press 'p' to see the lecture notes

Queries with Boolean operations

Negation of a comparison `"key" not ="value"`

```
{"key":{$not:{$eq:"value"}}}
```
not =

Negation of an expression `not (("key1"="value1") or ("key2"="value2"))`

```
{$nor:[{"key1":"value1"},{"key2":"value2"}]}
```
not or

Negation `nor ("key1"="value1")`

```
{$nor:[{"key1":"value1"}]}
```
not =

Queries with Boolean operations

Find all information about the suppliers who live in the `Netherlands` and have a contact title `Accounting Manager`

`$and`

```
db.orders.find({$and:[{"SUPPLIER.country":"Netherlands"},
                      {"SUPPLIER.contact title":"Accounting Manager"}]})
```

Find all information about the suppliers who live in the `Netherlands` or have a contact title `Accounting Manager`

`$or`

```
db.orders.find({$or:[{"SUPPLIER.country":"Netherlands"},
                     {"SUPPLIER.contact title":"Accounting Manager"}]})
```

Find all information about the customers who live in `France` or in `Germany`

`$in`

```
db.orders.find({"CUSTOMER.country":{$in:["France","Germany"]}})
```

Queries with Boolean operations

Find all information about the customers who do not live in `Germany`

$not

```
db.orders.find({"CUSTOMER.country":{$not:{$eq:"Germany"}}})
```

Find all information about the customers who do not both live in `Netherlands` or have a contact title `Accounting Manager`

Find all information about the customers who do not live in `Netherlands` and do not have a contact title `Accounting Manager`

$nor

```
db.orders.find({$nor:[{"SUPPLIER.country":"Netherlands"},
                      {"SUPPLIER.contact title":"Accounting Manager"}]})
```

Find all information about the customers who do not live in `Germany`

$nor

```
db.orders.find({$nor:[{"CUSTOMER.country":"Germany"}]})
```

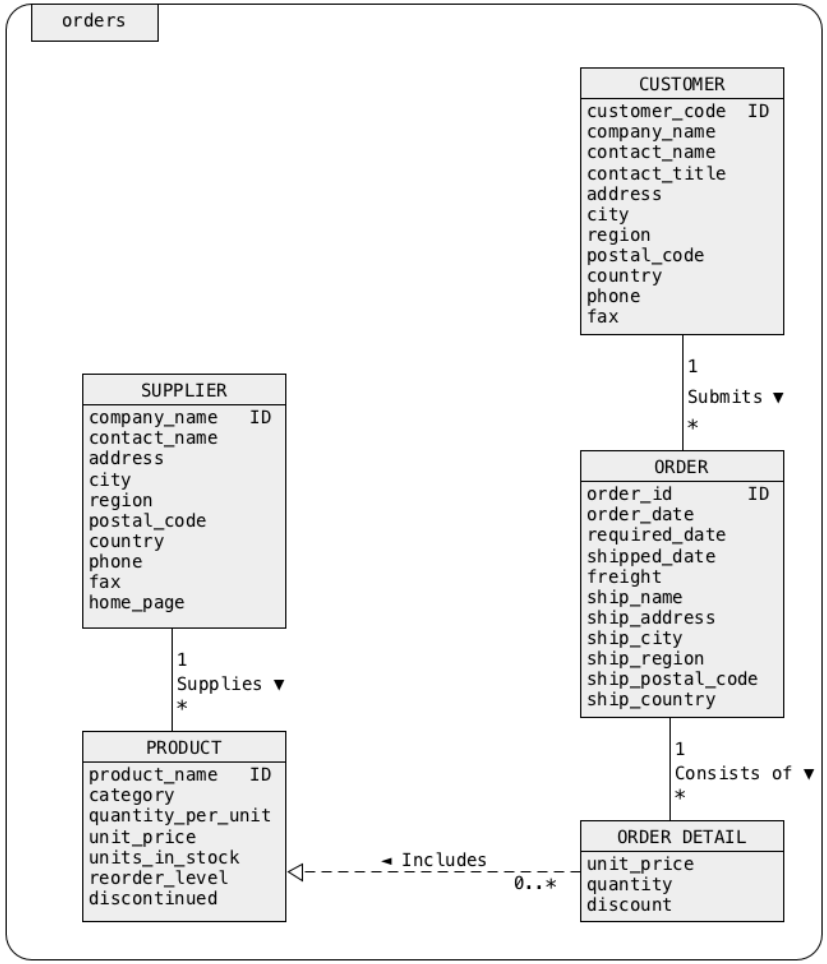
MongoDB Query language

Outline

MongoDB query language
A sample database
Simple queries
Queries with Boolean operations
Queries on nested documents
Queries on arrays
Projections
Queries about `NULL`s and missing keys
Iterations over a cursor

Queries on nested documents

A sample collection `orders`



In HTML view press 'p' to see the lecture notes
TOP

Queries on nested documents

Find all information about suppliers who supply a product named `Laughing Lumberjack Lager`

path

```
db.orders.find({"SUPPLIER.supplies.PRODUCT.product name":"Laughing Lumberjack Lager"})
```

Find all information about suppliers living in `London` who supply a product named `Chai`

path and path

```
db.orders.find({$and:[{"SUPPLIER.city":"London"},
                      {"SUPPLIER.supplies.PRODUCT.product name":"Chai"}]})
```

Find all information about suppliers living in `London` who supply a product named `Chai` or a product named `Chang`

(path or path) and path

```
db.orders.find({$and:[{"SUPPLIER.city":"London"},
                      {$or:[{"SUPPLIER.supplies.PRODUCT.product name":"Chai"},
                            {"SUPPLIER.supplies.PRODUCT.product name":"Chang"}]}]})
```

path and path

```
db.orders.find({$and:[{"SUPPLIER.city":"London"},
                      {"SUPPLIER.supplies.PRODUCT.product name":{$in:["Chai","Chang"]}}]})
```

Queries on nested documents

Find all information about suppliers living in `London` who supply a product named `Chai` and a product named `Chang`

path and path and path

```
db.orders.find({$and:[{"SUPPLIER.city":"London"},
                      {"SUPPLIER.supplies.PRODUCT.product name":"Chai"},
                      {"SUPPLIER.supplies.PRODUCT.product name":"Chang"}]})
```

Find all information about suppliers who supply at least one product

path and path

```
db.orders.find({$and:[{"SUPPLIER.supplies":{$exists:true}},
                      {"SUPPLIER.supplies":{$ne:[]}}]})
```

Find all information about suppliers who do not supply any products

path and path

```
db.orders.find({$and:[{"SUPPLIER.supplies":{$exists:true}},
                     {"SUPPLIER.supplies":{$eq:[]}}]})
```

Find all information about customers who submitted an order for at least one product `Flotemysost`

long path

```
db.orders.find({"CUSTOMER.submits.ORDER.consists of.ORDER DETAIL.product name":"Flotemysost"})
```

MongoDB Query language

Outline

MongoDB query language
A sample database
Simple queries
Queries with Boolean operations
Queries on nested documents
Queries on arrays
Projections
Queries about `NULL`s and missing keys
Iterations over a cursor

Queries on arrays

Array equal to `[1,2,3,4,5]`

```
{"array":{$all:[1,2,3,4,5]}}
```
find()

Array includes an element that satisfies a condition

```
{"array":{$elemMatch:{$eq:2}}}
```
find()

```
{"array":{$elemMatch:{$gt:2,$lt:4}}}
```
find()

Array includes a document satisfies a condition

```
{"array":{$elemMatch:{"key":{$eq:2}}}}
```
find()

```
{"array":{$elemMatch:{"key":{$gt:2,$lt:4}}}}
```
find()

Size of an array

```
{"array":{$size:5}}
```
find()

```
{"array":{$size:0}}
```
find()

```
{"array":[]}
```
find()

Queries on arrays

Find information about all customers who submitted an order that contains only a product `Boston Crab Meat` at unit price `14.7` in quantity `20` with discount equal to `0`

```
db.orders.find({"CUSTOMER.submits.ORDER.consists of":                                    = array
                                {$eq:[{"ORDER DETAIL":{"product name":"Boston Crab Meat",
                                                       "unit price":14.7,
                                                       "quantity":20,
                                                       "discount":0}}]}})
```

Find information about all suppliers such that their second supplied product is named `Chang`

```
                                                                                  path.n.path
db.orders.find({"SUPPLIER.supplies.1.PRODUCT.product name":"Chang"})
```

Find information about all suppliers such that their first supplied product is named `Chai` and the second supplied product is named `Chang`

```
                                                                    path.n.path and path.n.path
db.orders.find({$and:[{"SUPPLIER.supplies.0.PRODUCT.product name":"Chai"},
                      {"SUPPLIER.supplies.1.PRODUCT.product name":"Chang"}]})
```

Queries on arrays

Find information about all customers who purchased at least one product with discount greater than `0.2`

path > v

```
db.orders.find({"CUSTOMER.submits.ORDER.consists of.ORDER DETAIL.discount":{$gt:0.2}})
```

path &elemMatch v

```
db.orders.find({"CUSTOMER.submits.ORDER.consists of":{$elemMatch:{"ORDER DETAIL.discount":{$gt:0.2}}}})
```

Find information about all customers who purchased at least one product with discount equal to `0.25` and quantity equal to `16`

path &elemMatch v1, ...,vn

```
db.orders.find({"CUSTOMER.submits.ORDER.consists of":{$elemMatch:{"ORDER DETAIL.discount":{$eq:0.25},
                                                                  "ORDER DETAIL.quantity":{$eq:16}}}})
```

Find information about all customers who purchased 4 products in one order

$size

```
db.orders.find({"CUSTOMER.submits.ORDER.consists of":{$size:4}})
```

Queries on arrays

Find information about all customers who purchased more than 4 products in one order

path.n $exists

```
db.orders.find({"CUSTOMER.submits.ORDER.consists of.4":{$exists:true}})
```

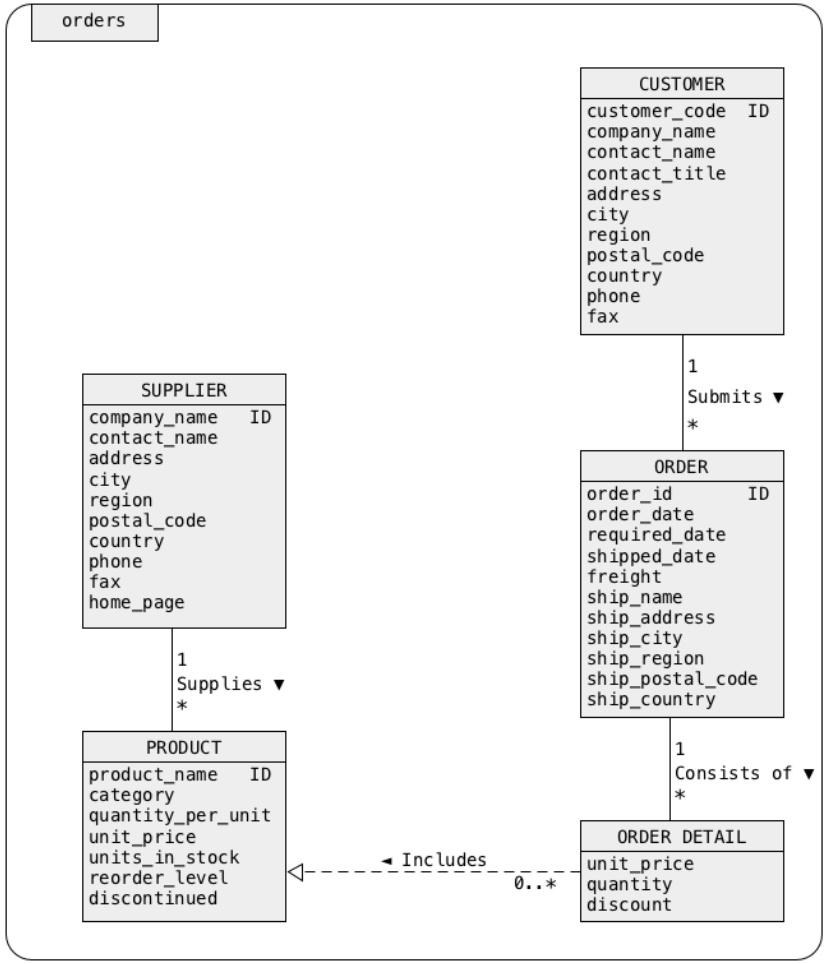
MongoDB Query language

Outline

- MongoDB query language
- A sample database
- Simple queries
- Queries with Boolean operations
- Queries on nested documents
- Queries on arrays
- Projections
- Queries about `NULL`s and missing keys
- Iterations over a cursor

Projections

A sample collection `orders`



In HTML view press 'p' to see the lecture notes

TOP

Projections

Find only a company name and contact name for all suppliers

Projection

```
db.orders.find({"SUPPLIER":{$exists:true}},{"_id":0,"SUPPLIER.company name":1,"SUPPLIER.contact name":1})
```

Find all information about suppliers except a company name, contact name and `_id`

Projection

```
db.orders.find({"SUPPLIER":{$exists:true}},{"_id":0,"SUPPLIER.company name":0,"SUPPLIER.contact name":0})
```

Find all information about suppliers except products supplied by suppliers and `_id`

Projection

```
db.orders.find({"SUPPLIER":{$exists:true}},{"_id":0,"SUPPLIER.supplies":0}))
```

Find only information about products supplied by suppliers

Projection

```
db.orders.find({"SUPPLIER":{$exists:true}},{"_id":0,"SUPPLIER.supplies":1}))
```

Projection

```
db.orders.find({"SUPPLIER":{$exists:true}},{"_id":0,"SUPPLIER.supplies.PRODUCT":1}))
```

Projections

Find only the names of products supplied by suppliers

Projection

```
db.orders.find({"SUPPLIER":{$exists:true}},{"_id":0,"SUPPLIER.supplies.PRODUCT.product name":1})
```

Find only the names of products and categories of products supplied by suppliers

Projection

```
db.orders.find({"SUPPLIER":{$exists:true}},{"_id":0,"SUPPLIER.supplies.PRODUCT.product name":1,
                                            "SUPPLIER.supplies.PRODUCT.category name":1})
```

Find only company names of suppliers and the names of products supplied by suppliers

Projection

```
db.orders.find({"SUPPLIER":{$exists:true}},{"_id":0,"SUPPLIER.company name":1,
                                            "SUPPLIER.supplies.PRODUCT.product name":1})
```

MongoDB Query language

Outline

- MongoDB query language
- A sample database
- Simple queries
- Queries with Boolean operations
- Queries on nested documents
- Queries on arrays
- Projections
- Queries about `NULL`s and missing keys
- Iterations over a cursor

Queries about nulls and missing keys

Find all information about the customers who have no `region` information

`null`

```
db.orders.find({"CUSTOMER.region":null})
```

Find all information about the customers who have `region` information

`$not null`

```
db.orders.find({"CUSTOMER.region":{$not:{$eq:null}}})
```

Find all information about the customers who have `PO Box` in their description

`$exists`

```
db.orders.find({"CUSTOMER.PO Box":{$exists:true}})
```

Find all information about the customers who do not have `PO Box` in their description

`$exists`

```
db.orders.find({"CUSTOMER.PO Box":{$exists:false}})
```

`$ not $exists`

```
db.orders.find({"CUSTOMER.PO Box":{$not:{$exists:true}}})
```

MongoDB Query language

Outline

MongoDB query language
A sample database
Simple queries
Queries with Boolean operations
Queries on nested documents
Queries on arrays
Projections
Queries about `NULL`s and missing keys
Iterations over a cursor

Iterations over a cursor

Create a cursor and display all information about suppliers

cursor

```
var cursor = db.orders.find({"SUPPLIER":{$exists:true}})
while(cursor.hasNext())
{ print(tojson(cursor.next())); }
```

cursor

```
var cursor = db.orders.find({"SUPPLIER":{$exists:true}})
cursor.forEach(printjson)
```

References

MongoDB Reference, Operators, Query and Projection Operators

Banker K., Bakkum P., Verch S., Garret D., Hawkins T., MongoDB in Action, 2nd ed., Manning Publishers, 2016