CSCI235 Database Systems

MongoDB Data Manipulations

Dr Janusz R. Getta

School of Computing and Information Technology -
University of Wollongong

MongoDB Data Manipluations

Outline

A sample database

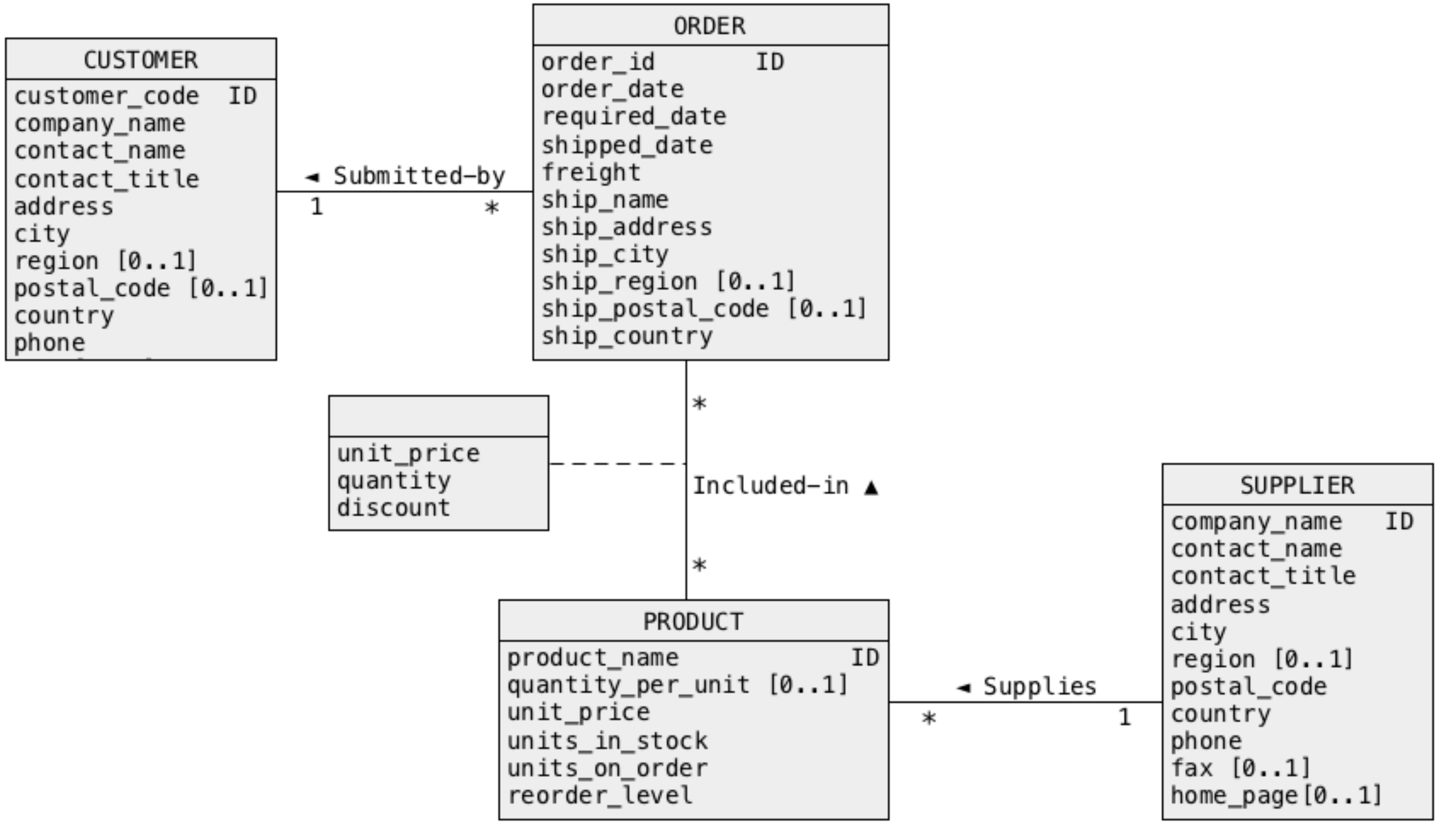
Inserting documents

Removing documents

Updating documents

A sample database

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



CUSTOMER

customer_code ID
company_name
contact_name
contact_title
address
city
region [0..1]
postal_code [0..1]
country
phone

◂ Submitted-by
1
*

ORDER

order_id ID
order_date
required_date
shipped_date
freight
ship_name
ship_address
ship_city
ship_region [0..1]
ship_postal_code [0..1]
ship_country

*
Included-in ▴
*

unit_price
quantity
discount

PRODUCT

product_name ID
quantity_per_unit [0..1]
unit_price
units_in_stock
units_on_order
reorder_level

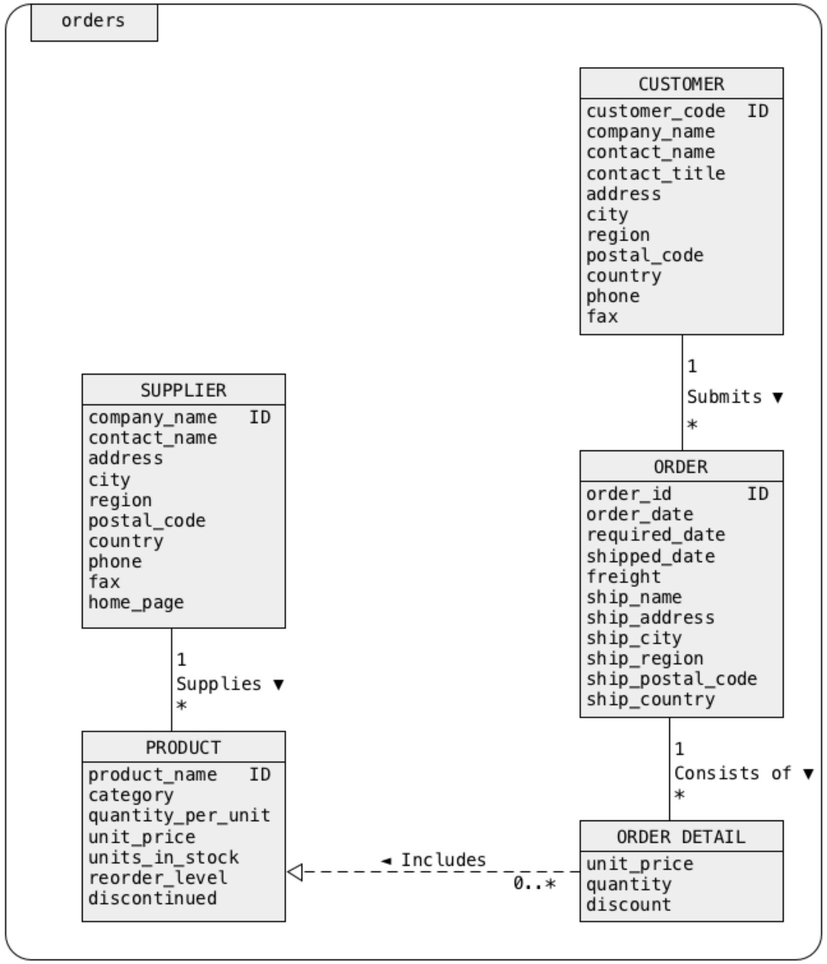
◂ Supplies
*
1

SUPPLIER

company_name ID
contact_name
contact_title
address
city
region [0..1]
postal_code
country
phone
fax [0..1]
home_page [0..1]

A sample database

A sample collection `orders`



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",
                                                                            ...  ...  ...  ...  ...
                                                                           }
                                                         }
                                                        ]
                                       }
                            }
 ] } }
```

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",
                                         ... ... ... ... ...
                                       }
                            }

                           ]
             }
}
```

MongoDB Data Manipluations

Outline

- A sample database
- Inserting documents
- Removing documents
- Updating documents

Inserting documents

A method `insert()` can be used to insert a single document into a collection

insert()

```
db.orders.insert(
      {
        "_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":[ ]
                     }
      }
)
```

Inserting documents

It is possible to insert many documents in one call

insert()

```
db.department.insert(
    [ {"_id":"ANATR",
       "CUSTOMER":{
                   "customer code":"ANATR",
                   "company name":"Ana Trujillo Emparedados y helados",
                   "contact name":"Ana Trujillo",
                   "contact title":"Owner",
                   "submits":[ ]
                  }
      },
      {"_id":"Aux joyeux ecclesiastiques",
       "SUPPLIER":{
                   "company name":"Aux joyeux ecclesiastiques",
                   "contact name":"Guylene Nodier",
                   "contact title":"Sales Manager",
                   "supplies": [ ]
                  }
      }
    ]
)
```

Inserting documents

Restrictions on insertions:

- It is impossible to insert two documents with the same `"_id"` into the same collection
- Insertion of many documents with one call can be done only on one collection
- There is a limit on much data can be inserted in one inserts (48Mb)
- If insertion of many documents with one call fails then entire operation is not rolled back (partial insertion)
- A document must be smaller than 16Mb

MongoDB Data Manipluations

Outline

A sample database

Inserting documents

Removing documents

Updating documents

Removing documents

A method `remove()` can be used to remove the documents from a collection

Remove all documents from a collection `orders`

remove()

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

Remove all customers from a collection `orders`

remove()

```
db.orders.remove({"CUSTOMER":{$exists:true}})
```

Remove from a collection `orders` all suppliers who supply no products

remove()

```
db.orders.remove({"SUPPLIER.supplies":[]})
```

Drop a collection `orders`

drop()

```
db.orders.drop()
```

MongoDB: Data Manipluationss

Outline

- A sample database
- Inserting documents
- Removing documents
- Updating documents

Updating documents

Replace a document that describes a customer with customer code `ALFKI` with a new document

update()

```
db.orders.update({"CUSTOMER.customer code":"ALFKI"},
                 {"_id":"ALFKI",
                  "CUSTOMER":{"customer code" :"ALFKI",
                              "company name":"Alfreds Futterkiste",
                              "contact name":"James Bond",
                              "contact title":"Sales Representative",
                              "address":"Johan Strauss Str. 23",
                              "city":"Hamburg",
                              "region":null,
                              "postal code":"34567",
                              "country":"Germany",
                              "phone":"030-0074321",
                              "fax":"030-0076545",
                              "submits":[ ]
                             }
                 }
)
```

Updating documents

Rename a key `contact title` into `title` in all documents describing customers

update()

```
db.orders.updateMany({"CUSTOMER":{$exists:true}},
                     {$rename:{"CUSTOMER.contact title":"CUSTOMER.title"}})
```

Change a name of city to `Liverpool` in a document describing a customer who has customer code `AROUT`

update()

```
db.orders.updateOne({"CUSTOMER.customer code":"AROUT"},
                    {$set:{"CUSTOMER.city":"Liverpool"}})
```

Change a name of city to `null` in a document describing a customer who has a customer code `AROUT`

update()

```
db.orders.updateOne({"CUSTOMER.customer code":"AROUT"},
                    {$set:{"CUSTOMER.city":null}})
```

Remove a key city from a document describing a customer who has a customer code `AROUT`

update()

```
db.orders.updateOne({"CUSTOMER.customer code":"AROUT"},
                    {$unset:{"CUSTOMER.city":null}}))
```

Updating documents

Change a unit price of a product `Guarana Fantastica` to `10`

update()

```
db.orders.updateOne({"SUPPLIER.supplies.PRODUCT.product name":"Guarana Fantastica"},
                    {$set:{"SUPPLIER.supplies.$.PRODUCT.unit price":100}}))
```

Set a value of a unit price of a product `Guarana Fantastica` to `null`

update()

```
db.orders.updateOne({"SUPPLIER.supplies.PRODUCT.product name":"Guarana Fantastica"},
                    {$set:{"SUPPLIER.supplies.$.PRODUCT.unit price":null}}))
```

Increment a unit price of a product `Guarana Fantastica` by `10`

update()

```
db.orders.updateOne({"SUPPLIER.supplies.PRODUCT.product name":"Guarana Fantastica"},
                    {$inc:{"SUPPLIER.supplies.$.PRODUCT.unit price":10}})
```

Multiply a unit price of a product `Guarana Fantastica` by `2`

update()

```
 db.orders.updateOne({"SUPPLIER.supplies.PRODUCT.product name":"Guarana Fantastica"},
                     {$mul:{"SUPPLIER.supplies.$.PRODUCT.unit price":2}})
```

Updating documents

Remove a key unit price from a description of a product `Guarana Fantastica`

update()

```
db.orders.updateOne({"SUPPLIER.supplies.PRODUCT.product name":"Guarana Fantastica"},
                    {$unset:{"SUPPLIER.supplies.$.PRODUCT.unit price":null}}))
```

Multiply a unit price of the first product supplied by a company `Exotic Liquids` by `2`

update()

```
db.orders.updateOne({"SUPPLIER.company name":"Exotic Liquids"},
                    {$mul:{"SUPPLIER.supplies.0.PRODUCT.unit price":2}})
```

If we know, that the first product supplied by a company `Exotic Liquids` is `Chai` then its unit price can be multiplied by `2` in the following way

update()

```
db.orders.updateOne({"SUPPLIER.supplies.PRODUCT.product name":"Chai"},
                    {$mul:{"SUPPLIER.supplies.$.PRODUCT.unit price":2}})
```

Updating documents

Remove a product `Chai` supplied by a supplier `Exotic Liquids`

update()

```
db.orders.updateOne({"SUPPLIER.company name":"Exotic Liquids"},
                    {$pull:{"SUPPLIER.supplies":{"PRODUCT.product name":"Chai"}}})
```

Remove all discontinued products supplied by a supplier `Bigfoot Breweries`

update()

```
db.orders.updateMany({"SUPPLIER.company name":"Bigfoot Breweries"},
                     {$pull:{"SUPPLIER.supplies":{"PRODUCT.discontinued":"Y"}}}))
```

Add a new product `Champagne` to the products supplied by `Exotic Liquids`, the product is not discontinued and its unit price is `100`

update()

```
db.orders.updateOne({"SUPPLIER.company name":"Exotic Liquids"},
                    {$push:{"SUPPLIER.supplies":{"PRODUCT":{"product name":"Champaigne",
                                                            "discontinued":"N",
                                                            "unit price":100}}}}))
```

References

MongoDB Reference, Operators, Update Operators

Banker L., Bakkum P., Verch S., Garrett D., Hawkins T., MongoDB in Action, Manning, 2016

Chodorow K. MongoDB The Definitive Guide, O'Reilly, 2013, chapter 2