CSCI835 Database Systems

BSON Design

Dr Janusz R. Getta

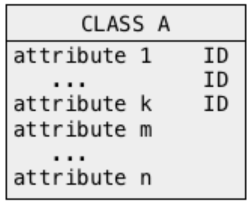
School of Computing and Information Technology -
University of Wollongong

BSON Design

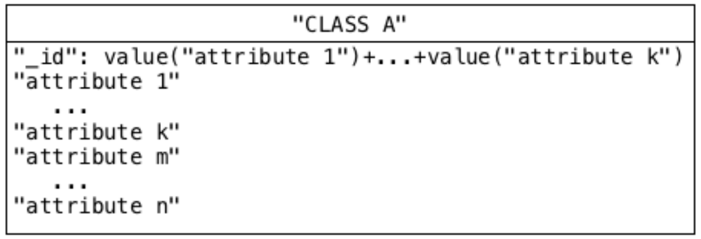
Outline

Class

Conceptual schema



Logical schema



Class

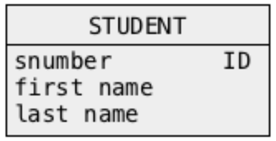
JSON Schema

$jsonSchema validator

```
db.createCollection("class_a",
                    { "validator":{$jsonSchema:
  {"bsonType":"object",
   "properties":{"_id":{"bsonType":"string"},
                 "CLASS A":{"bsonType":"object",
                            "properties":{"attribute 1":{"bsonType": ... },
                                          ...            ...          ...  ,
                                          "attribute k":{"bsonType": ... },
                                          "attribute m":{"bsonType": ... },
                                          ...            ...          ...  ,
                                          "attribute n":{"bsonType": ... } },
                            "required":["attribute 1",...,"attribute k","attribute m",...,"attribute n"],
                            "additionalProperties":false }
                },
   "required":["_id","CLASS_A"],
   "additionalProperties":false
  } } } );
```

Class

Example



$jsonSchema validator

```
db.createCollection("student",
                    { "validator":{$jsonSchema:
  {"bsonType":"object",
   "properties":{"_id":{"bsonType":"string"},
                 "STUDENT":{"bsonType":"object",
                            "properties":{"snumber":{"bsonType":"int" },
                                          "first name":{"bsonType":"string" },
                                          "last name":{"bsonType":"string" } },
                            "required":["snumber","first name","last name"],
                            "additionalProperties":false }
                },
   "required":["_id","STUDENT"],
   "additionalProperties":false
  } } } );
```

Class

Example

STUDENT

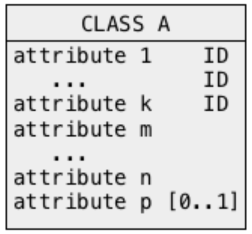
```
db.student.insert({"_id":"1234567",
                   "STUDENT":{"snumber":NumberInt("1234567"),
                              "first name":"Harry",
                              "last name":"Potter"}
                  } );
```

BSON Design

Outline

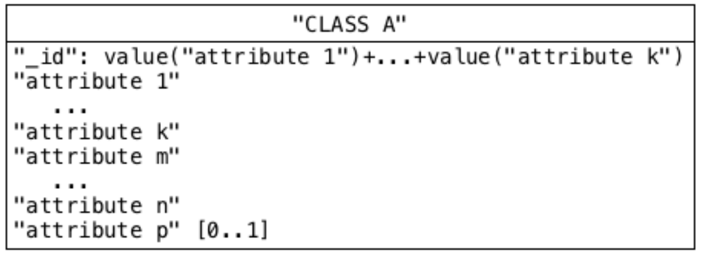
Optional attribute

Conceptual schema



```
        CLASS A
attribute 1     ID
   ...          ID
attribute k     ID
attribute m
   ...
attribute n
attribute p [0..1]
```

Logical schema



```
                         "CLASS A"
"_id": value("attribute 1")+...+value("attribute k")
"attribute 1"
   ...
"attribute k"
"attribute m"
   ...
"attribute n"
"attribute p"  [0..1]
```

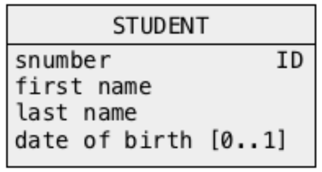
Optional attribute

JSON Schema

```
                                                              $jsonSchema validator
db.createCollection("class_a",
                    { "validator":{$jsonSchema:
  {"bsonType":"object",
   "properties":{"_id":{"bsonType":"string"},
                 "CLASS A":{"bsonType":"object",
                            "properties":{"attribute 1":{"bsonType": ... },
                                          ...             ...          ...  ,
                                          "attribute k":{"bsonType": ... },
                                          "attribute m":{"bsonType": ... },
                                          ...             ...          ...  ,
                                          "attribute n":{"bsonType": ... },
                                          "attribute p":{"bsonType": ... } },
                            "required":["attribute 1",...,"attribute k","attribute m",...,"attribute n"],
                            "additionalProperties":false }
                },
   "required":["_id","CLASS_A"],
   "additionalProperties":false
  } } } );
```

Optional attribute

Example



$jsonSchema validator

```
db.createCollection("student",
                    { "validator":{$jsonSchema:
  {"bsonType":"object",
   "properties":{"_id":{"bsonType":"string"},
                 "STUDENT":{"bsonType":"object",
                            "properties":{"snumber":{"bsonType":"int" },
                                          "first name":{"bsonType":"string" },
                                          "last name":{"bsonType":"string" },
                                          "date of birth":{"bsonType":"date"} },
                            "required":["snumber","first name","last name"],
                            "additionalProperties":false }
                },
   "required":["_id","STUDENT"],
   "additionalProperties":false
  } } } );
```

Optional attribute

Example

STUDENT

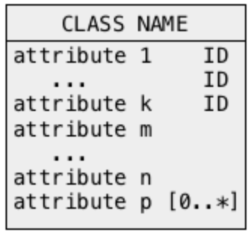
```
db.student.insert({"_id":"1234567",
                   "STUDENT":{"snumber":NumberInt("1234567"),
                              "first name":"Harry",
                              "last name":"Potter",
                              "date of birth":Date("1999-07-07")},
                  } );
```

BSON Design

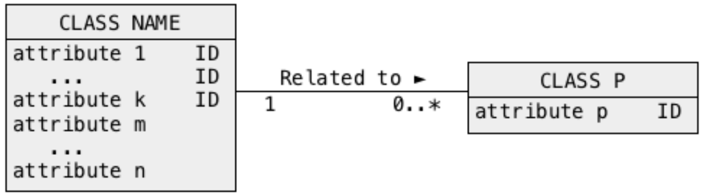
Outline

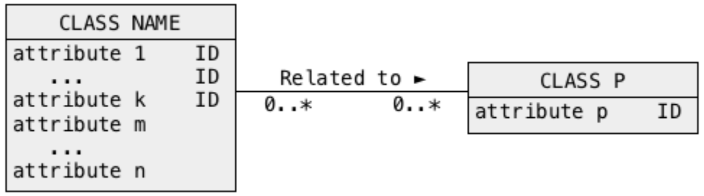
Multivalued attribute

Conceptual schema



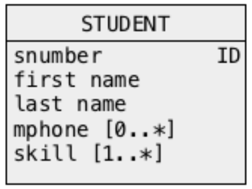
Equivalent conceptual schemas



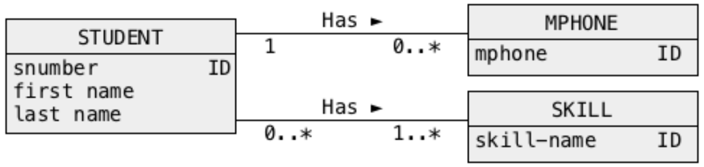


Multivalued attribute

Sample conceptual schema



Equivalent conceptual schema

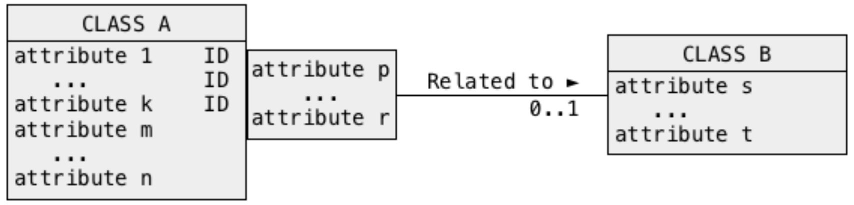


BSON Design

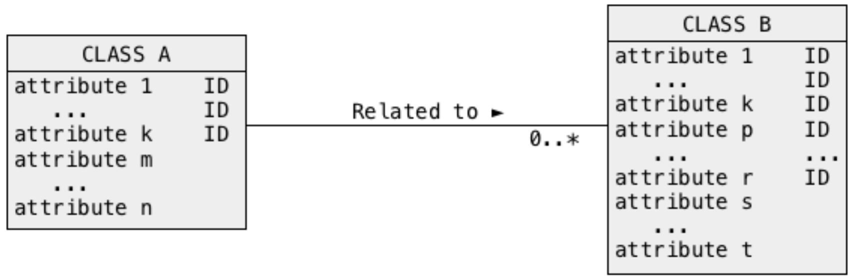
Outline

Qualification

Conceptual schema

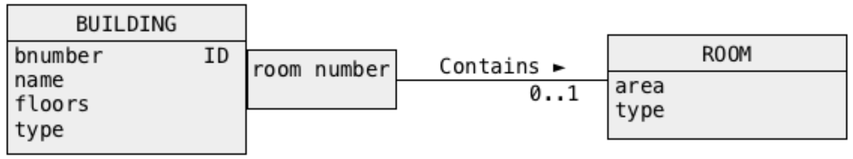


Equivalent conceptual schema

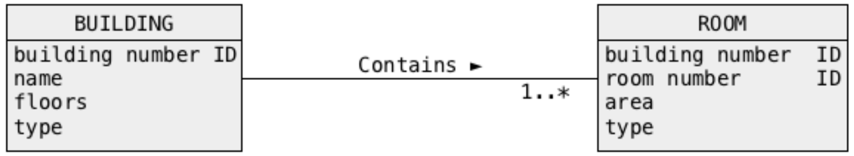


Qualification

Sample conceptual schema



Equivalent conceptual schema

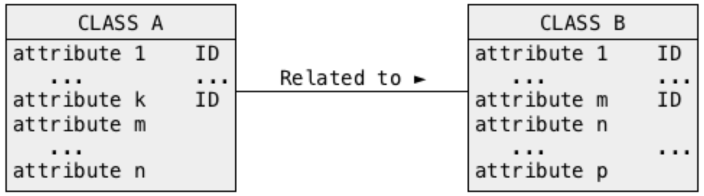


BSON Design

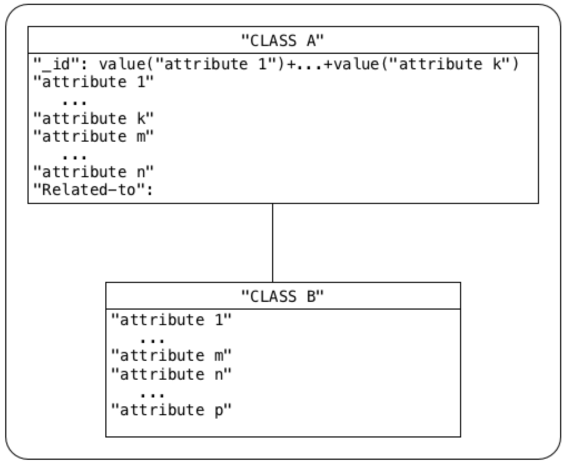
Outline

One-to-one association

Conceptual schema

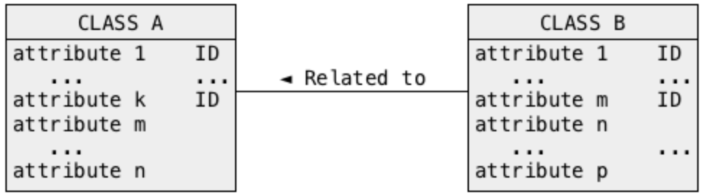


Logical schema

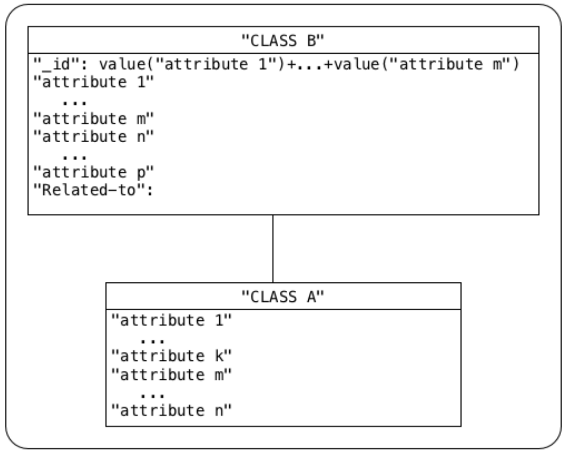


One-to-one association

Conceptual schema



Logical schema



One-to-one association

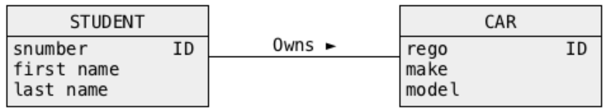
JSON Schema

$jsonSchema validator

```
db.createCollection("class_a",
                    { "validator":{$jsonSchema:
{"bsonType":"object"
 "properties":{"_id":{"bsonType":"string"},
               "CLASS A":{"bsonType":"object",
                          "properties":{"attribute 1":{"bsonType": ... },
                                        ...            ...         ...  ,
                                        "attribute k":{"bsonType": ... },
                                        "attribute m":{"bsonType": ... },
                                        ...            ...         ...  ,
                                        "attribute n":{"bsonType": ... },
                                        "Related-to": {"bsonType":"object",
                                                       "properties":{"CLASS B":{"bsonType":"object",
                                                                                "properties":{"attribute 1":{"bsonType": ... },
                                                                                              ...            ...         ...  ,
                                                                                              "attribute m":{"bsonType": ... },
                                                                                              "attribute n":{"bsonType": ... },
                                                                                              ...                        ...  ,
                                                                                              "attribute p":{"bsonType": ... } },
                                                                                "required":["attribute 1",...,"attribute m","attribute n",...,"attribute p"],
                                                                                "additionalProperties":false } },
                                                       "required":["CLASS B"],
                                                       "additionalProperties":false } },
                          "required":["attribute 1",...,"attribute k","attribute m",...,"attribute n","Related-to"],
                          "additionalProperties":false} },
 "required":["_id","CLASS_A"],
 "additionalProperties":false
} } } );
```

One-to-one association

Example



$jsonSchema validator

```
db.createCollection("student",
                    { "validator":{$jsonSchema:
  {"bsonType":"object",
   "properties":{"_id":{"bsonType":"string"},
                 "STUDENT":{"bsonType":"object",
                            "properties":{"snumber":{"bsonType":"int"},
                                          "first name":{"bsonType":"string"},
                                          "last name":{"bsonType":"string"},
                                          "Owns":{"bsonType":"object",
                                                  "properties":{"CAR":{"bsonType":"object",
                                                                       "properties":{"rego":{"bsonType":"string"},
                                                                                     "make":{"bsonType":"string"},
                                                                                     "model":{"bsonType":"string"} },
                                                                       "required":["rego","make","model"],
                                                                       "additionalProperties":false} },
                                                  "required":["CAR"],
                                                  "additionalProperties":false} },
                             "required":["snumber","first name","last name","Owns"],
                             "additionalProperties":false} },
   "required":["_id","STUDENT"],
   "additionalProperties":false
  } } } );
```

One-to-one association

Example

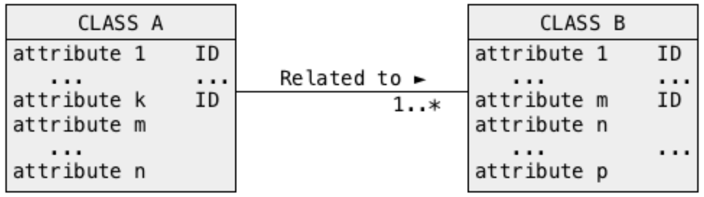
```
                                                                    STUDENT Owns Car
db.student.insert({"_id":"1234567",
                   "STUDENT":{"snumber":NumberInt("1234567"),
                              "first name":"Harry",
                              "last name":"Potter",
                              "Owns":{"CAR":{"rego":"AL08UK",
                                             "make":"Rolls Royce",
                                             "model":"Silver Shadow"}
                                     }
                             }
                  } );
```

BSON Design

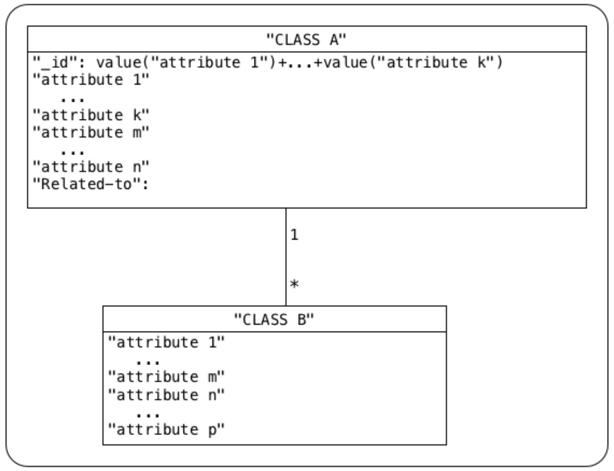
Outline

One-to-many association

Conceptual schema



Logical schema



One-to-many association

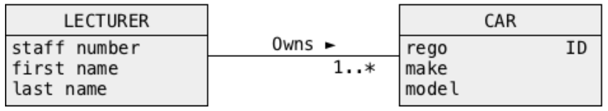
JSON Schema

$jsonSchema validator

```
db.createCollection("class_a",
                    { "validator":{$jsonSchema:
{"bsonType":"object"
 "properties":{"_id":{"bsonType":"string"},
               "CLASS A":{"bsonType":"object",
                          "properties":{"attribute 1":{"bsonType": ... },
                                        ...            ...         ...  ,
                                        "attribute k":{"bsonType": ... },
                                        "attribute m":{"bsonType": ... },
                                        ...            ...         ...  ,
                                        "attribute n":{"bsonType": ... },
                                        "Related-to": {"bsonType":"array",
                                                       "items":{"bsonType":"object",
                                                                "properties":{"CLASS B":{"bsonType":"object",
                                                                                         "properties":{"attribute 1":{"bsonType": ... },
                                                                                                       ...            ...         ...  ,
                                                                                                       "attribute m":{"bsonType": ... },
                                                                                                       "attribute n":{"bsonType": ... },
                                                                                                       ...                        ...  ,
                                                                                                       "attribute p":{"bsonType": ... } },
                                                                           "required":["attribute 1",...,"attribute m","attribute n",...,"attribute p"],
                                                                                         "additionalProperties":false } },
                                                                "required":["CLASS B"],
                                                                "additionalProperties":false } },
                          "required":["attribute 1",...,"attribute k","attribute m",...,"attribute n","Related-to"],
                          "additionalProperties":false} },
 "required":["_id","CLASS_A"],
 "additionalProperties":false } } );
```

One-to-many association

Example



$jsonSchema validator

```
db.createCollection("lecturer",
                    { "validator":{$jsonSchema:
  {"bsonType":"object",
   "properties":{"_id":{"bsonType":"string"},
                 "LECTURER":{"bsonType":"object",
                             "properties":{"staff number":{"bsonType":"int"},
                                           "first name":{"bsonType":"string"},
                                           "last name":{"bsonType":"string"},
                                           "Owns":{"bsonType":"array",
                                                   "items":{"bsonType":"object",
                                                            "properties":{"CAR":{"bsonType":"object",
                                                                                 "properties":{"rego":{"bsonType":"string"},
                                                                                               "make":{"bsonType":"string"},
                                                                                               "model":{"bsonType":"string"} },
                                                                                 "required":["rego","make","model"],
                                                                                 "additionalProperties":false} },
                                                            "required":["CAR"],
                                                            "additionalProperties":false} } },
                              "required":["snumber","first name","last name","Owns"],
                              "additionalProperties":false} },
   "required":["_id","LECTURER"],
   "additionalProperties":false
  } } } );
```

One-to-many association

Example

LECTURER Owns CAR

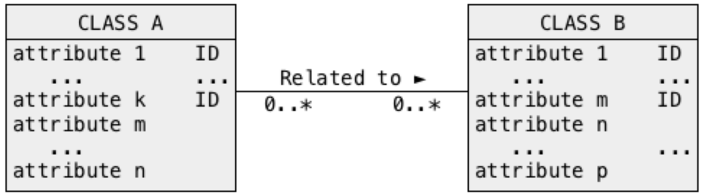
```
db.lecturer.insert({"_id":"007",
                   "LECTURER":{"staff number":NumberInt("007"),
                               "first name":"James",
                               "last name":"Bond",
                               "Owns":[{"CAR":{"rego":"AL08UK",
                                               "make":"Rolls Royce",
                                               "model":"Silver Shadow"} },
                                       {"CAR":{"rego":"PKR856",
                                               "make":"Mercedes",
                                               "model";"SE800"} } ]
                              }
                   } );
```

BSON Design

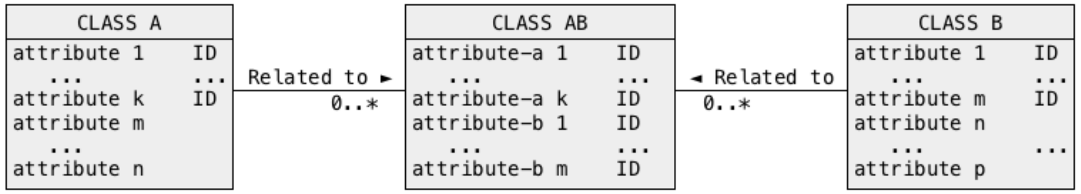
Outline

Many-to-many association

Conceptual schema

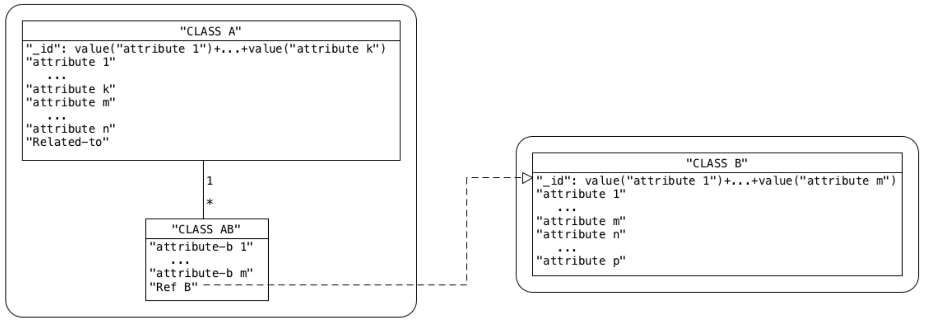


Equivalent conceptual schema

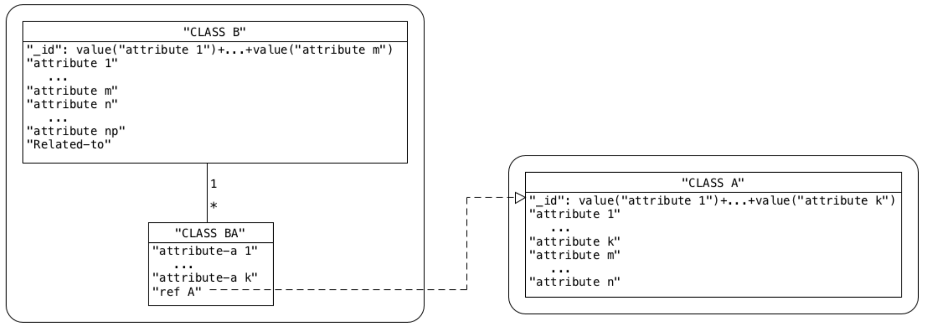


Many-to-many association

Conceptual schema



Equivalent conceptual schema



Many-to-many association

JSON Schema

$jsonSchema validator

```
db.createCollection("class_a",
                    { "validator":{$jsonSchema:
{"bsonType":"object"
 "properties":{"_id":{"bsonType":"string"},
               "CLASS A":{"bsonType":"object",
                          "properties":{"attribute 1":{"bsonType": ... },
                                        ...            ...         ...  ,
                                        "attribute k":{"bsonType": ... },
                                        "attribute m":{"bsonType": ... },
                                        ...            ...         ...  ,
                                        "attribute n":{"bsonType": ... },
                                        "Related-to": {"bsonType":"array",
                                                       "items":{"bsonType":"object",
                                                                "properties":{"CLASS AB":{"bsonType":"object",
                                                                                          "properties":{"attribute-b 1":{"bsonType": ... },
                                                                                                        ...            ...         ...  ,
                                                                                                        "attribute-b m":{"bsonType": ... },
                                                                                                        "Ref B"        :{"bsonType": ... } },
                                                                           "required":["attribute-b 1",...,"attribute-b m","Ref B"],
                                                                                         "additionalProperties":false } },
                                                                "required":["CLASS AB"],
                                                                "additionalProperties":false } },
                          "required":["attribute 1",...,"attribute k","attribute m",...,"attribute n","Related-to"],
                          "additionalProperties":false} },
 "required":["_id","CLASS_A"],
 "additionalProperties":false } } } );
```

Many-to-many association

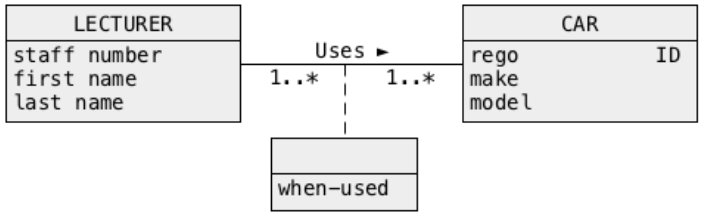
JSON Schema

$jsonSchema validator

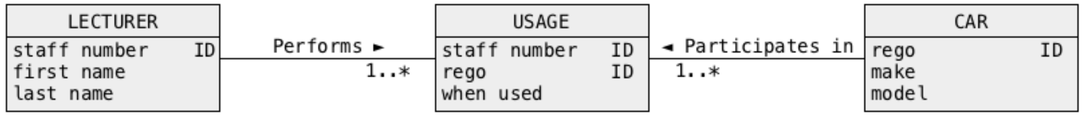
```
db.createCollection("class_b",
                    { "validator":{$jsonSchema:
{"bsonType":"object"
 "properties":{"_id":{"bsonType":"string"},
               "CLASS B":{"bsonType":"object",
                          "properties":{"attribute 1":{"bsonType": ... },
                                        ...         ...        ...  ,
                                        "attribute m":{"bsonType": ... },
                                        "attribute n":{"bsonType": ... },
                                        ...         ...        ...  ,
                                        "attribute p":{"bsonType": ... } },
                          "required":["attribute 1",...,"attribute m","attribute n",...,"attribute p"],
                          "additionalProperties":false} },
 "required":["_id","CLASS_B"],
 "additionalProperties":false } } } );
```

Many-to-many association

Example



Equivalent conceptual schema



Many-to-many association

Example

$jsonSchema validator

```
db.createCollection("lecturer",
                    { "validator":{$jsonSchema:
{"bsonType":"object"
 "properties":{"_id":{"bsonType":"string"},
              "LECTURER":{"bsonType":"object",
                          "properties":{"staff number":{"bsonType":"string" },
                                       "first name":{"bsonType":"string"},
                                       "last name":{"bsonType":"string" },
                                       "Performs": {"bsonType":"array",
                                                   "items":{"bsonType":"object",
                                                           "properties":{"USAGE":{"bsonType":"object",
                                                                                  "properties":{"rego":{"bsonType":"string"},
                                                                                               "when used":{"bsonType":"date"} },
                                                                                  "required":["rego","when used"],
                                                                                  "additionalProperties":false } },
                                                             "required":["USAGE"],
                                                             "additionalProperties":false } },
                         "required":["staff number","first name","last name","Performs"],
                         "additionalProperties":false} },
 "required":["_id","LECTURER"],
 "additionalProperties":false } } } );
```

Class

Example

$jsonSchema validator

```
db.createCollection("car",
                    { "validator":{$jsonSchema:
  {"bsonType":"object",
   "properties":{"_id":{"bsonType":"string"},
                 "CAR":{"bsonType":"object",
                        "properties":{"rego":{"bsonType":"string" },
                                      "make":{"bsonType":"string" },
                                      "model":{"bsonType":"string" } },
                            "required":["rego","make","model"],
                            "additionalProperties":false }
                },
   "required":["_id","CAR"],
   "additionalProperties":false
  } } } );
```

One-to-many association

Example

```
                                                                    LECTURER Uses CAR
db.lecturer.insert({"_id":"007",
                   "LECTURER":{"staff number":NumberInt("007"),
                              "first name":"James",
                              "last name":"Bond",
                              "Performs":[ {"USAGE":{"rego":"AL08UK",
                                                     "when used":Date("2017-07-08")} },
                                           {"USAGE":{"rego":"PKR856",
                                                     "when used":Date("2017-07-09")} },
                                           {"USAGE":{"rego":"AL08UK",
                                                     "when used":Date("2017-07-09")} } ]
                              }
} );
```

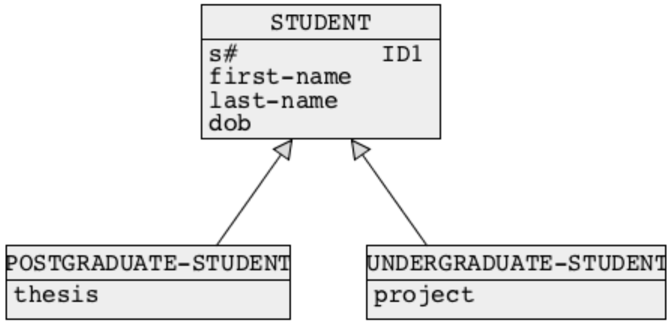
```
                                                                    LECTURER Uses CAR
db.car.insert({"_id":"AL08UK",
                   "CAR":{"rego":"AL08UK","make":"Honda","model":"Legend"}
} );
```

```
                                                                    LECTURER Uses CAR
db.car.insert({"_id":"PKR856",
                   "CAR":{"rego":"PKR856","make":"Rolls Royce","model":"Silver Shadow"}
} );
```

BSON Design

Outline

Generalizations - superset method

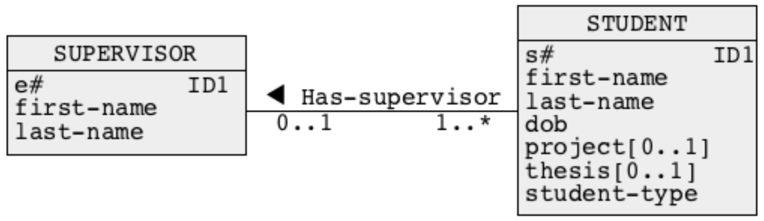


A superset method transforms entire generalization hierarchy into a single class of objects in the following way:

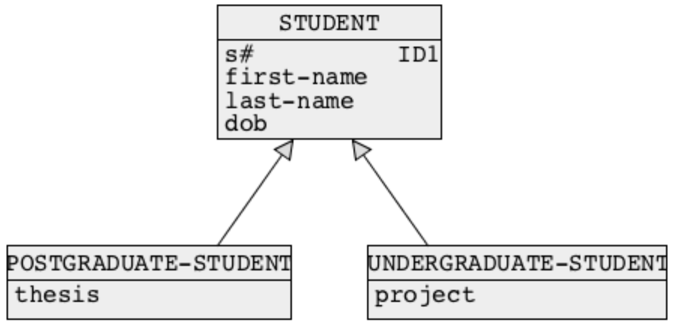
- All attributes from the classes of objects at the lowest level of generalization hierarchy are copied to an immediate higher level and become optional attributes ([0..1] tag) there, e.g. the attributes `project` and `thesis` are copied from the classes `UNDERGRADUATE-STUDENT` and `POSTGRADUATE-STUDENT` to a class `STUDENT`
- An attribute `type-of-superclass` is added to a superclass, e.g. and attribute `type-of-students` is added to a class `STUDENT`

Generalizations - superset method

- All classes at the lowest level are removed
- The steps above are repeated until only one class of objects is left



Generalizations - subset method

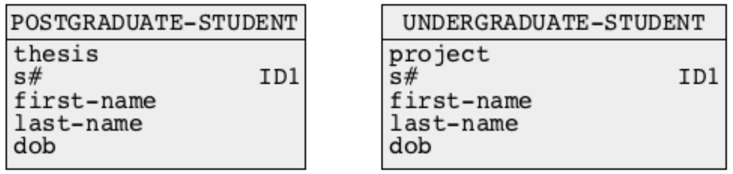


A subset method transforms entire generalization hierarchy into a number of classes of objects in the following way:

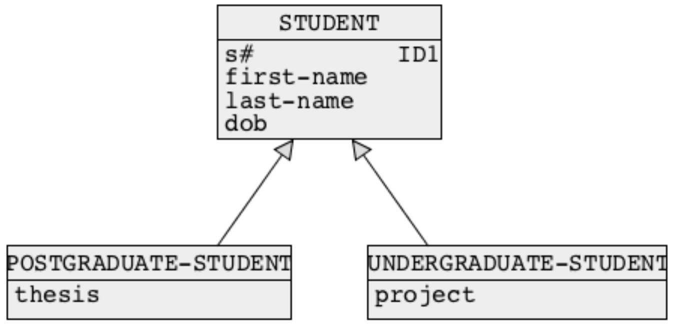
- All attributes from the classes of objects at the higher levels of generalization hierarchy are copied to the classes of objects at the lowest levels of generalization hierarchy e.g. the attributes `s#` and `first-name` last-name, dob are copied from a class `STUDENT` to the classes `POSTGRADUATE-STUDENT` and `UNDERGRADUATE-STUDENT`

Generalizations - subset method

- All classes of objects except those at the lowest levels of generalization hierachy are removed, e.g. a class `STUDENT` is removed



Generalizations - association method

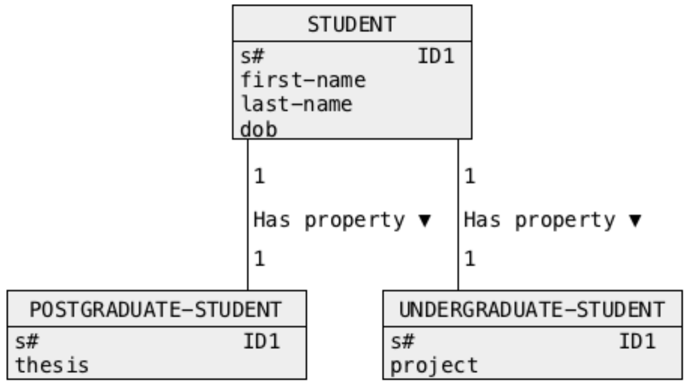


An association method transforms entire generalization hierarchy into a number of classes of objects in the following way:

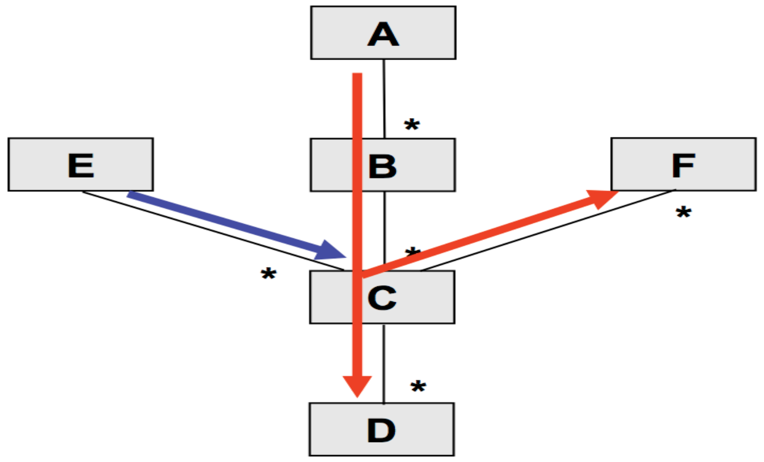
- One of the identifiers from a superclass is copied to subclasses one level below a superclass, e.g. an attribute `s#` is copied from a class `STUDENT` to the classes `UNDEGRADUATE-STUDENT and POSTGRADUATE-STUDENT`

Generalizations - association method

- A generalization level is removed from a diagram



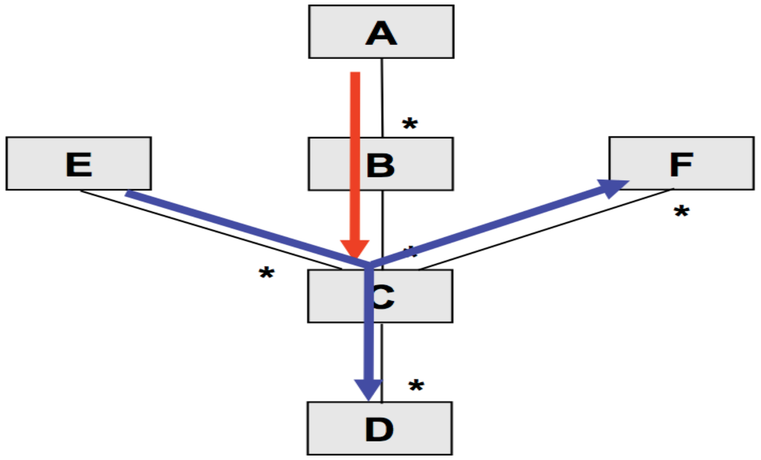
Representation of network structures



Implementation of sample hierarchy

```
"A"
   ...
   "B"
      ...
      "C":{"REF":"e"},
           ...
           "D"...
               "F"...
"E":{"_id":"e", ... }
```

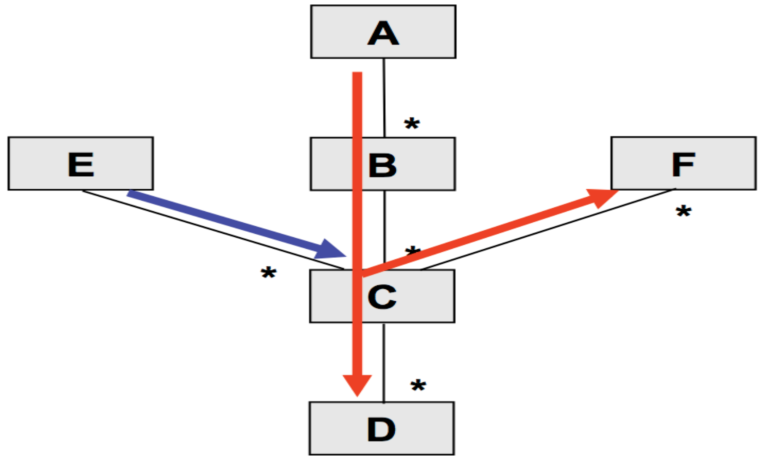
Representation of network structures



Implementation of sample hierarchy

```
"E"
    ...
    "C":{"REF":"b"}
         ...
         "D"
         "F"
"A"
   ...
   "B":{"ID":"b", ... }
```

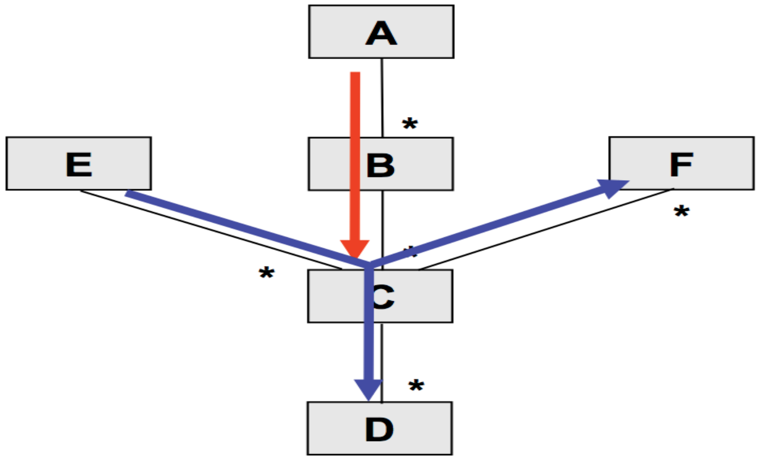
Representation of network structures



Implementation of sample hierarchy

```
"A"
   ...
   "B"
      ...
      "C":{"ID":"c"},
           ...
           "D"...
           "F"...
"E":[{"REF":"c"}, ...]
```

Representation of network structures



Implementation of sample hierarchy

```
"E"
    ...
    "C":{"ID":"c"}
         ...
         "D"
         "F"
"A"
   ...
   "B":[{"REF":"c"}, ... ]
```

References

JSON Schema

Understanding JSON Schema

MongoDB - JSON Schema validation

MongoDB - $jsonSchema operator