**Лабораторная работа №7**

**Создание и использование ссылок**

Нереляционные СУБД позволяют проектировать модель предметной области в виде набора объектов. При этом информация об одной сущности, разбросанная по различным таблицам РБД, в нереляциионной БД будет собрана в одном объекте. Главным отличие БД mongo от РБД является отсутствие аналога операции соединения (JOIN). Если существует необходимость использовать соединения в базе данных, то они реализуются в программном коде приложения. Для того чтобы найти данные, связанные с каким-либо документом, как правило, необходимо выполнить второй запрос.

Для связывания документов можно сохранять их вместе с «\_id» связанных документов.

В качестве примера проиллюстрируем сохранение информации о производителе телефонов в виде связанной записи.

Далее следует выполнять команды, выделенные **жирным шрифтом.**

Создадим новую коллекцию

**> db.createCollection("Model")**

Добавим в нее документ

**> db.Model.insert({\_id: ObjectId ("000000000000000000000001"), "Name": "Nokia", "BrandName": "Nokia", "BrandCountry": "Finland"})**

На документ с именем «Nokia» будут ссылаться другие документы. Для создания связанного документа необходимо знать поле «\_id» документа «Nokia». Запись с указанием фирмы производителя будет выглядеть следующим образом:

**> db.Model.insert( {\_id: ObjectId ("000000000000000000000002"), "Name": "L920", "Model": "Lumia 920", "OSFamily": "Windows", "OSVersion": "8", "Brand": ObjectId ("000000000000000000000001")})**

Обратите внимание, что значение поля «Brand» документа с именем «L920» и поля «\_id» документа «Nokia» совпадают. Поле «\_id» может быть любым уникальным значением. Чтобы найти все телефоны, произведенные под брендом «Nokia», необходимо выполнить запрос с указанием значения его поля «\_id»:

**> db.Model.find ({Brand: ObjectId ("000000000000000000000001")})**

В результате получим

{ "\_id" : ObjectId("000000000000000000000002"), "Name" : "L920", "Model" : "Lumia 920", "OSFamily" : "Windows", "OSVersion" : "8", "Brand" : ObjectId("000000000000000000000001") }

Введем еще один документ, значение поля «Brand» которого также совпадает с полем «\_id» документа «Nokia»

**>db.Model.insert({\_id: ObjectId ("000000000000000000000003"), "Name":"Samsung","Model":"Galaxy S3","OSFamily":"Android","OSVersion":"4.0 ice Cream Sandwich","Brand": ObjectId ("000000000000000000000001")})**

Тогда в результате выполненного запроса

**> db.Model.find ({Brand: ObjectId ("000000000000000000000001")})**

получим оба связанных с документом «Nokia» документа

{ "\_id" : ObjectId("000000000000000000000002"), "Name" : "L920", "Model" : "Lumia 920", "OSFamily" : "Windows", "OSVersion" : "8", "Brand" : ObjectId("000000000000000000000001") }

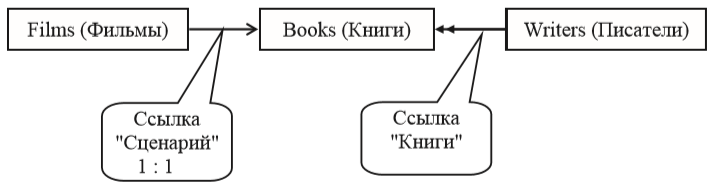
{ "\_id" : ObjectId("000000000000000000000003"), "Name" : "Samsung", "Model" : "Galaxy S3", "OSFamily" : "Android", "OSVersion" : "4.0 ice Cream Sandwich", "Brand" : ObjectId("000000000000000000000001") }

Возможность создавать ненормализованные документы сложной структуры, содержащие вложенные документы и/или их массивы, позволяет адекватно представлять в базе отдельные объекты, имеющие сложную структуру. Для представления взаимодействий между объектами используются ссылки. Ссылка — атрибут в документе, значение которого используется для доступа к другому документу. Ссылки являются средством прямого представления связей и используются для навигационной обработки документов .

Рассмотрим «ручные» ссылки. Для однозначной идентификации документа в MongoDB ручная ссылка содержит три атрибута: $db:<Имя базы>, $ref:<Имя коллекции>, $id:<Ключ документа>.

В атрибуте $db указывается имя базы, которая содержит ссылочный документ. При обращении к документу ссылка $db необязательна. При ее отсутствии поиск документа ведется в установленной базе. Атрибуты $ref и $id определяют коллекцию и идентификатор ссылочного документа в коллекции. Для обращения к документу по хранящейся в БД ссылке необходимо сначала найти исходный документ и извлечь атрибуты ссылки, а затем использовать их в новом запросе к ссылочному документу.

Пусть создана БД, которая содержит информацию о кинофильмах в коллекции Films (Фильмы), данные о писателях (сценаристах) в коллекции Writers (Писатели), а сведения об их произведениях размещены в документах коллекции Books (Книги). Связи данных показаны на рис. . В документе о фильме предусмотрен атрибут-ссылка "Сценарий", указывающий на книгу, по которой поставлен фильм. В документе, представляющем автора, предусмотрен массив ссылок на книги, написанные этим автором.



Рассмотрим способ создания и использования ручных ссылок на примере ссылки "Сценарий". Пусть коллекция Films содержит информацию о фильме "Ночной дозор":

**> db.createCollection("Films")**

**> db.Films.insert({"\_id": ObjectId ("55c8214cb7f9469326e345cb"), "Название": "Ночной дозор", "Режиссер": "Бекмамбетов Т. Н.", "Год": 2004})**

В коллекцию Books помещаем сведения о романе "Ночной дозор", на основе которого был снят фильм

**> db.createCollection("Books")**

**> db.Books.insert({"\_id": ObjectId ("55c82635b7f9469326e345cc"), "Название": "Ночной дозор", "Год": 1998, "Тип": "Фантастика"})**

В соответствии со схемой данных (рис. ) для представления сведений о книге в записи о фильме следует создать ссылку на роман "Ночной дозор". Добавим в фильм "Ночной дозор" атрибут "Сценарий", содержащий ссылку на роман "Ночной дозор" в коллекции Books.

**> db.Films.update ({"Название":"Ночной дозор"}, {$set:{"Сценарий":{"$ref": "Books", "$id": ObjectId ("55c82635b7f9469326e345cc")}}});**

Проверка добавленной ссылки вернет измененную запись о фильме:

**> db.Films.find ()**

{ "\_id" : ObjectId("55c8214cb7f9469326e345cb"), "Название" : "Ночной дозор", "Режиссер" : "Бекмамбетов Т. Н.", "Год" : 2004, "Сценарий" : DBRef("Books", ObjectId("55c82635b7f9469326e345cc")) }

Обратите внимание: значение ссылки в атрибуте "Сценарий" теперь представлено функцией DBRef (…).

Для проверки ссылки найдем роман, по которому поставлен фильм "Ночной дозор". Поиск выполняется в два этапа.

1. Сначала надо получить ссылку. Для этого извлекаем из документа "Ночной дозор" атрибут "Сценарий", содержащий ссылку на книгу и сохраняем в переменной "Ссылка":

**> var Ссылка = db.Films.findOne ({"Название": "Ночной дозор"}, {"Сценарий": 1, \_id: 0});**

Получаем в переменной "Ссылка": {"Сценарий":

D B R e f ( " B o o k s " , O b j e c t I d ("55c82635b7f9469326e345cc"))}

2. Затем выбираем книгу по хранящейся в переменной ссылке:

**> db [Ссылка.Сценарий.$ref].findOne ({"\_id": (Ссылка.Сценарий.$id)});**

**{**

Перед выполнением команда findOne модифицируется подстановкой из атрибута-ссылки $ref имени коллекции, а из атрибута $id — ключа документа таким образом, что выполняется метод: db.Books.findOne ({"\_id": ObjectId

("55c82635b7f9469326e345cc")}), который возвращает сведения о книге:

"\_id" : ObjectId("55c82635b7f9469326e345cc"),

"Название" : "Ночной дозор",

"Год" : 1998,

"Тип" : "Фантастика"

}

**Задание для самостоятельной работы**

1, Создать базу данных, включающую документы, содержащие анкетные данные студентов Вашей группы (фамилия и инициалы, пол, возраст, телефон, семейное положение) и документы, содержащие информацию о результатах сдачи последней экзаменационной сессии каждым студентом группы.

2.Связать документы, содержащие информацию об одном и том же студенте

**Вопросы для защиты работы**

1. Для чего нужны ссылки?
2. Какие атрибуты содержит ручная ссылка ?
3. Что определяет атрибут $db?
4. Что определяет атрибут $ref ?
5. Что определяет атрибут и $id?

**Содержание отчета**

1.Номер и название лабораторной работы

2. Экранные формы, показывающие порядок выполнения лабораторной

работы, и результаты, полученные в ходе её выполнения.

3.Постановка задачи для самостоятельной работы

4. Экранные формы, показывающие порядок выполнения задания для

Самостоятельной работы с соответствующими пояснениями, и результаты, полученные в ходе её выполнения.

5.Ответы на вопросы для защиты