Cleansing Time – SQL Free Applications

Cleansing Time - SQL Free Applications

Release 3.4

MongoDB, Inc.

May 30, 2017

Contents

1 RDBMS to MongoDB Workshop 2
- 1.1 RDBMS to MongoDB Introduction . . . 2
- 1.2 System Requirements . . . 3
- 1.3 Environment Setup . . . 4
- 1.4 MongoMart RDBMS . . . 6
- 1.5 Migration Strategies . . . 9
- 1.6 Lab1: `Reviews` Migration . . . 11
- 1.7 Lab2: `Items` Migration . . . 13

1 RDBMS to MongoDB Workshop

RDBMS to MongoDB Introduction **(page 2)** Activities to get the workshop started

System Requirements **(page 3)** Let's review our system requirements to run the workshop

Environment Setup **(page 4)** MongoMart supported by RDBMS backend

MongoMart RDBMS **(page 6)** MongoMart supported by RDBMS backend

Migration Strategies **(page 9)** Different Database migration strategies review.

Lab1: Reviews Migration **(page 11)** Migrate Reviews data to MongoDB

Lab2: Items Migration **(page 13)** Migrate Items data to MongoDB

<*no title*> (page ??) Migrate Cart data to MongoDB

<*no title*> (page ??) Migrate Stores data to MongoDB

<*no title*> (page ??) MongoMart fully migrated to MongoDB

1.1 RDBMS to MongoDB Introduction

Welcome

This is a full day workshop where we are going to be covering the following topics:

- Migration strategies
- Architecture and application implications
- Relation to Document modeling
- MongoDB CRUD operations

Where do we start?

You will be given a fully functional application called **Mongomart** backed by a relational database.

You will be tasked with defining the different tasks required to move this application to a MongoDB supported back-end.

After each task the instructor will provide a solution for each of the labs/tasks.

How are we going to do that?

Given a migration strategy, defined by the instructor, we will migrating our application.

We need to understand:

- What's the relation schema looking like. *(ERD)*
- How do we want to store the same information in a MongoDB document model.

What to expect by the end of the workshop?

By the of this workshop you will be more suited to:

- Understand the practical tasks required to move a relational system to MongoDB
- The benefits and tradeoffs of the different migration approaches
- Considerations about schema design and relational to document mapping

1.2 System Requirements

Before you get started

Before we jump into coding and making a migration plan let's review the list of software components required to run this workshop.

In this workshop we will be using a set of software reqirements apart from the actual code and workbooks.

Mongomart Java Version

- Java 8[1]
- Apache Maven[2]

Code has been tested using Java 8[3] and built using Apache Maven[4] 3.5.0 .

Other versions may function correctly but we cannot provide efficient support.

[1] http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html
[2] https://maven.apache.org/install.html
[3] http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html
[4] https://maven.apache.org/install.html

Mongomart Databases

- MongoDB 3.4[5]
- MySQL[6]

We will be using MySQL[7] Ver 5.7.18 and MongoDB 3.4[8].

System Editor or IDE

During the course of the workshop we will be requiring some code changes and recompilations.

Make sure to have your prefered editor or java IDE.

Otherwise take some time to download and install Eclipse[9] or Intellij[10].

1.3 Environment Setup

Setup Workshop Environment

After we completed the download of our workshop material, it is then time to bring our **MongoMart** up.

To do so, we will need the following:

- Unzip `mongomart.zip` file
- Launch local MySQL server
- Import dataset
- Run `mongomart` process

Unzip `mongomart.zip`

After inflating the `mongomart.zip` file, we will find this directory structure:

```
unzip mongomart.zip
ls mongomart
> README  dataset java
```

[5] https://docs.mongodb.com/manual/installation/
[6] https://dev.mysql.com/downloads/installer/
[7] https://dev.mysql.com/downloads/installer/
[8] https://docs.mongodb.com/manual/installation/
[9] http://www.eclipse.org/downloads/eclipse-packages/?show_instructions=TRUE
[10] https://www.jetbrains.com/idea/download

Launch local MySQL server

Time to launch our relational database server:

- In your *NIX system

```
mysql.server start
```

- Or Windows

```
C:\> "C:\Program Files\MySQL\MySQL Server 5.7\bin\mysqld"
```

Import dataset

Within your `mongomart` folder, there is a `dataset` folder.

This folder contains the dataset that we will be working with.

To import this dataset:

```
# creates the relational system schema
mysql -uroot < dataset/create_schema.sql
# imports previously generated dump
mysql -uroot < dataset/dump/mongomart.sql
# run a few checks
mysql -uroot < dataset/check.sql
```

And for our Windows friends:

```
cmd.exe /c "mysql -u root < dataset\create_schema.sql"
cmd.exe /c "mysql -u root < dataset\dump\mongomart.sql"
cmd.exe /c "mysql -u root < dataset\mongomart.sql"
```

Run the `mongomart` app

Once we have our dataset fully imported it is time for us to give `mongmart` a spin:

- First we generate the java package

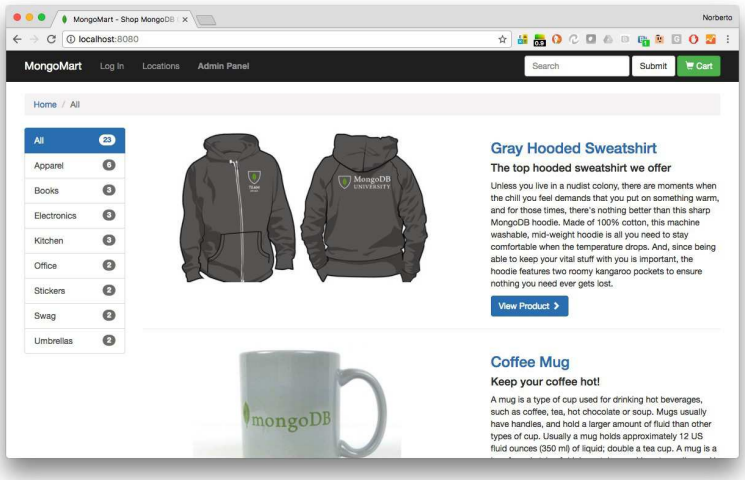
```
cd mongomart/java
mvn package -Dmaven.test.skip=true
```

- Then we run the application process

```
java -jar target/MongoMart-1.0-SNAPSHOT.jar
```

Final Step

Once the process is correctly up and running, the final step is to connect to http://localhost:8080 using your system browser.



1.4 MongoMart RDBMS

What is MongoMart

MongoMart is an on-line store for buying MongoDB merchandise. In this workshop we will start with a RDBMS version and end up with a MongoDB one.

MongoMart Demo of Fully Implemented Version

- View Items
- View Items by Category
- Text Search
- View Item Details
- Shopping Cart

View Items

- http://localhost:8080
- Pagination and page numbers
- Click on a category

View Items by Category

- http://localhost:8080/?category=Apparel
- Pagination and page numbers
- “All” is listed as a category, to return to all items listing

Text Search

- http://localhost:8080/search?query=shirt
- This functionality is not yet implemented.
- Will be part of this workshop to add Text Search functionality

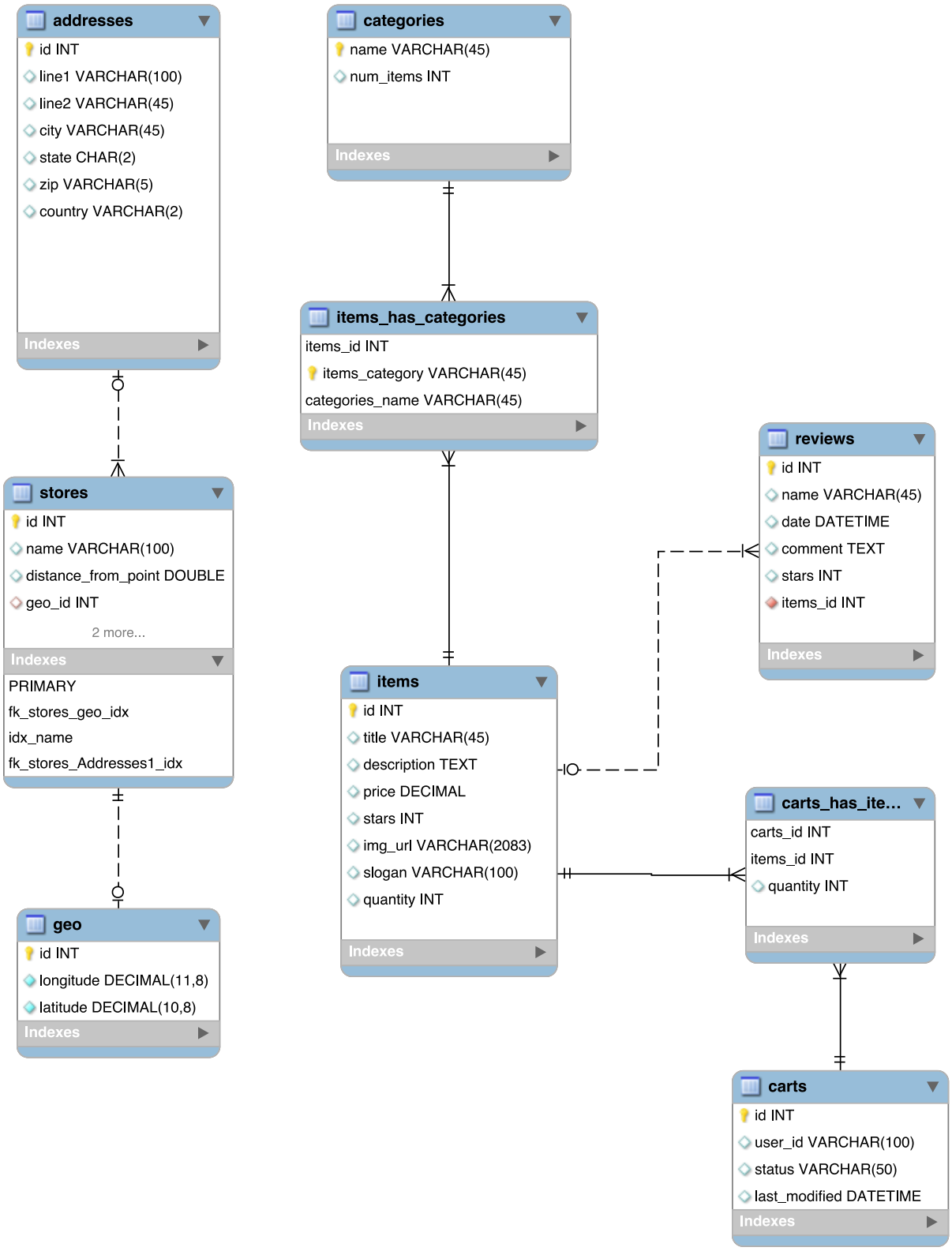
View Item Details

- http://localhost:8080/item?id=1
- Star rating based on reviews
- Add a review
- Related items
- Add item to cart

Shopping Cart

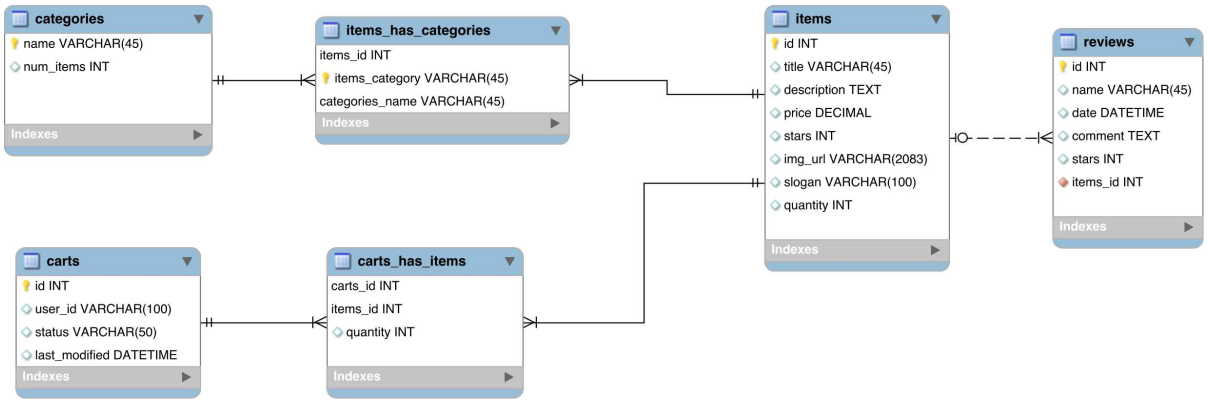
- http://localhost:8080/cart
- Adding an item multiple times increments quantity by 1
- Change quantity of any item
- Changing quantity to 0 removes item

Entity Relationship Diagram (ERD)



Stores ERD

Items ERD



1.5 Migration Strategies

Learning Objectives

In this module we will be covering the following topics:

- Different strategies to deal with database migration
- Different development implications of such strategies
- Pros and cons of different strategies and approaches

To extend your understanding about this topic, we recommend you to read our Migration Guide White Paper[11] that explores this topic in detail.

[11] https://www.mongodb.com/collateral/rdbms-mongodb-migration-guide

Migration Strategies

When migrating between databases, there are two typical strategies that we can follow:

- Full one stop shop migration
- Step-by-step hybrid migration

Database Migration Phases

But regardless of the strategy/approach taken, there are a few steps or phases that we need to attend:

- Schema Design: Plan how our schema is going to look like in the new system
- App Integration: What different libraries and CRUD operations will need to be modified.
- Data Migration: The actual migration of our data between databases
- Operations: All the backup/monitoring/management procedures will require a review.

Schema Design

The way that data is organized in a RDBMS system is considerably different from the one used by MongoDB.

Having that in mind, we need to understand how restructure our data structures to make the most out-of the new system that we are migrating to..

App Integration

// TODO

Operations

// TODO

Full one stop shop migration

// TODO

Step-by-step hybrid migration

// TODO

1.6 Lab1: `Reviews` Migration

Learning Objectives

In this lab, we'll be setting our application into an hybrid mode where both databases will be providing service.

We will be covering:

- Benefits of using an hybrid solution for migration purposes.
- Implementation strategies to perform such migration.
- Schema design review.

Introduction

Currently we have our `ReviewsController` using only one RDBMS system.

For this lab we purpose that `Reviews` should be migrated to MongoDB.

To do that we will change the `ReviewsController` to use both RDBMS and MognoDB.

Strategy

To do this, the data migration we will follow this approach:

- For each `item`, we will requests all `review` stored in MongoDB **mongomart.review** collection.
- Then, we will request all `reviews` from the RDBMS system that are not present in MongoDB
- For each `review` retrived from RDBMS we will store a copy `review` in MongoDB
- All new reviews added to the system should be stored in MongoDB

`review` Collection Schema Design

In this collection we will be storing data in a format that reflects the `Reviews` class.

```
{
  "_id": <ObjectId>,
  "id": <integer>,
  "name": <string>,
  "date" <ISODate>,
  "comment": <string>,
  "stars": <integer>,
  "items_id": <integer>
}
```

Step 1: Connection to MongoDB

In order for us to be able to store data into MongoDB we will have to establish a connection from our application to our server:

- Bring up a MongoDB server instance

```
# default dbpath for MongoDB
mkdir -p /data/db
# launch MongoDB
mongod
```

Once the instance is up and running lets establish a connection from our application.

- Change the `MongoMart` class to include a connection to MongoDB.

Step 2: Create `mongodb.ReviewDao`

To interact with MongoDB collection we will need to create a `ReviewsDao` class.

- Create a new package `/mongomart/dao/mongodb`
- Create the `ReviewsDao` class within this new package
  - `mongomart.dao.mongodb.ReviewsDao`
- This dao should reimplement all rdbms package public methods:
  - `getItemReviews(...)`
  - `avgStars(...)`
  - `numReviews(...)`

Step 3: Add `addReview` method

As part of our startegy, we will need to add reviews to the `review` collection.

Our `dao.mongodb.ReviewsDao` class should have a method that adds reviews to the collection

- Implement an `addReview()` method

Step 4: Integrate new Dao into `StoreController`

After we've added our new `dao.mongodb.ReviewsDao` we will need to start using it.

The `reviews` table is access by this `StoreController`. The correspoding MongoDB collection, will also be accessed from the same class.

- `StoreController` constructor shoukld receive a MongoDB database object
- Use the new `dao.mongodb.ReviewsDao` to find reviews in MongoDB
- After collecting all reviews, insert those reviews into MongoDB.

Add Reviews to Items

At this point you should have a fully functional **MongoMart** that stores all new reviews on `Item` into MongoDB.

Test this by selecting a few random items and add reviews into them.

To fully migrate, we could run script that iterates over all items.

This will use the internal application code to perform the migration!

1.7 Lab2: `Items` Migration

Learning Objectives

In this lab we will be exploring the following operations:

- Enforce a full migration using hybrid solution
- Enable Text Search

Step 1: Add `dao.mongodb.ItemDao` to MongoMart

After receiving this new `ItemDao` class you will have to:

- Integrate this class within the **MongoMart** application
  - `StoreController` is a good candidate for us to into!
- Once this new Dao is integrated
  - Validate that `Items` are getting stored into MongoDB

Step 2: Iterate over all `Item`

To migrate all items into MongoDB, we can simply run the following script:

```
# iterate over all 23 items
for i in {1..23}
do
  curl -I http://localhost:8080/item?id=$i
done
```

... and for our Windows friends:

```
FOR /L %%A IN (1,1,23) DO( curl -I http://localhost:8080/item?id=%%A )
```

Step 3: Enable Text Search

At this point, the Text Search[12] functionality is not yet enabled. Time to get it up and running:

- Go ahead and enable Text Search[13] on `items` collection.
- This functionality is provided by MongoDB
  - Text Search[14] should be enabled in the following fields
    * `title`
    * `slogan`
    * `description`

// TODO need to finish this one

// TODO need to finish this one

// TODO need to finish this one

[12] https://docs.mongodb.com/manual/reference/operator/query/text/
[13] https://docs.mongodb.com/manual/reference/operator/query/text/
[14] https://docs.mongodb.com/manual/reference/operator/query/text/

Find out more
mongodb.com | mongodb.org
university.mongodb.com

Having trouble?
File a JIRA ticket:
jira.mongodb.org

Follow us on twitter
@MongoDBInc
@MongoDB