**Faculty of Computing, Engineering and Media Coursework Brief 2019/20**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Module name:** | | **Database Management and Programming** | | | | | | |
| **Module code:** | | **IMAT3104** | | | | | | |
| **Title of the Assignment:** | | **NoSQL Coursework** | | | | | | |
| **This coursework item is:** (delete as appropriate) | | | | **Summative** | | |  | |
| **This summative coursework will be marked anonymously:**  (delete as appropriate) | | | | | |  | | **No** |
| **The learning outcomes that are assessed by this coursework are:**  1. produce a data model that accurately reflects a complex business scenario  3. program a NoSQL database to store, manipulate, and retrieve data and perform aggregation functions. | | | | | | | | |
| **This coursework is:** (delete as appropriate) | | | Individual | | | |  | |
| If other or mixed ... explain here: N/A | | | | | | | | |
| **This coursework constitutes** 50 % **of the overall module mark.** | | | | | | | | |
| **Date Set:** | 19th February 2021 | | | | | | | |
| **Date & Time Due:** | **Friday 26th March 2021, 13:00** | | | | | | | |
| **Your marked coursework and feedback will be available to you on:**  If for any reason this is not forthcoming by the due date your module leader will let you know why and when it can be expected. The Associate Professor Student Experience ([cemstudentexperience@dmu.ac.uk)](mailto:cemstudentexperience@dmu.ac.uk) should be informed of any issues relating to the return of marked coursework and feedback.  Note that you should normally receive feedback on your coursework by **no later than 20 University working days after the formal hand-in date,** provided that you have met the submission deadline. | | | | | 21 working days after submission | | | |
| **When completed you are required to submit your coursework via:**  1. Upload electronic copy to Blackboard via TurnItIn  **If you need any support or advice on completing this coursework please visit the Student Matters tab on the Faculty of Technology Blackboard page.** | | | | | | | | |
| **Late submission of coursework policy:** Late submissions will be processed in accordance with current University regulations which state:  *“the time period during which a student may submit a piece of work late without authorisation and have the work capped at 40% [50% at PG level] if passed is* ***14 calendar days****. Work submitted unauthorised more than 14 calendar days after the original submission date will receive a mark of 0%. These regulations apply to a student’s first attempt at coursework. Work submitted late without authorisation which constitutes reassessment of a previously failed piece of coursework will always receive a mark of 0%.”* | | | | | | | | |
| **Academic Offences and Bad Academic Practices:**  These include plagiarism, cheating, collusion, copying work and reuse of your own work, poor referencing or the passing off of somebody else's ideas as your own. If you are in any doubt about what constitutes an academic offence or bad academic practice you must check with your tutor.  Further information and details of how DSU can support you, if needed, is available at:  [http://www.dmu.ac.uk/dmu-students/the-student-gateway/academic-support-office/academic-](http://www.dmu.ac.uk/dmu-students/the-student-gateway/academic-support-office/academic-offences.aspx) [offences.aspx](http://www.dmu.ac.uk/dmu-students/the-student-gateway/academic-support-office/academic-offences.aspx) and | | | | | | | | |

|  |  |
| --- | --- |
| [http://www.dmu.ac.uk/dmu-students/the-student-gateway/academic-support-office/bad-academic-](http://www.dmu.ac.uk/dmu-students/the-student-gateway/academic-support-office/bad-academic-practice.aspx) [practice.aspx](http://www.dmu.ac.uk/dmu-students/the-student-gateway/academic-support-office/bad-academic-practice.aspx) | |
| **Tasks to be undertaken:**  Here is a summary of the tasks for this coursework:   1. Query the collections (queries Q1 to Q9). 2. Implement and Explain Index for the database (E1). 3. Re-design the database (D1).   Read the full specification enclosed below for the details. | |
| **Deliverables to be submitted for assessment:**  You are required to upload your answers to the questions into Turnitin on Blackboard as one file (PDF). All questions and answers must be labelled with their capital letter and number (i.e. Q1 to Q9, E1 and D1). Add your P number to your filename. Your file must be in a readable format so the NoSQL code in plain text (Q1 to Q9, E1 and D1) can be copied and executed from it (in DOC or PDF format).  Read the full specification enclosed below for the details. | |
| **How the work will be marked:**  Mark totals are shown alongside each question. Your tutor will need to able to execute your NoSQL code in MongoDB to check for correctness.  Marks will be awarded for correctness and presentation. Marks will typically be deducted if information is erroneous, missing, irrelevant or difficult to ascertain. Marks can also be deducted if the answer is particularly inefficient. Therefore, partial marks are available if an answer is partially correct or partially presented.  There is often more than one way to answer a question and so it is possible to gain full marks for an answer even if it is different from the markers' specimen set of answers. However, if these answers do not follow the examples and exercises taught on the module they may be inferior in some way and so consequently marks may be deducted. Note that some questions hint at the most appropriate method to use.  Read the full specification enclosed below for the details. | |
| **Module leader/tutor name:** | **Hakeem Ibrahim** |
| **Contact details:** | hakeem.ibrahim@dmu.ac.uk |

# IMAT3104 Database Management and Programming NoSQL Reassessment Coursework 2020-2021

**Introduction**

This is an individual assignment, which gives you an opportunity to demonstrate your knowledge of NoSQL and your ability to implement, query and design a MongoDB document database. You will be awarded marks for what is achieved. This assignment is worth 50% of the overall module mark. The deadline is Friday 26th March 2020, 1pm.

# Scenario

Amazon.com, commonly known as Amazon, is an American e-commerce and cloud computing company that was founded in 1994. It is the largest internet-based retailer in the world by total sales and market capitalization. It sells an enormous variety of products and stores huge datasets which includes reviews (ratings, text, helpfulness votes etc.) and product metadata (descriptions, category information, price, brand, and image features etc.).

Amazon has a web page for each product, e.g. <https://www.amazon.com/dp/B00005BHKS> and one for each reviewer, e.g.

<http://www.amazon.com/gp/cdp/member-reviews/A1RVTBD2QZ8TQ8>.

A subset of Amazon data has been selected for this coursework. The product and review data is based on some clothing, shoes and jewellery spanning October 1999 to July 2014.

Permission was granted to use the data for this coursework from the following source: <http://jmcauley.ucsd.edu/data/amazon/>on condition that the authors’ research papers are referenced (He & McAuley 2016), (McAuley *et al.* 2015).

# Personalise and Load data into collections

Download the two zipped files from Blackboard; they are called:

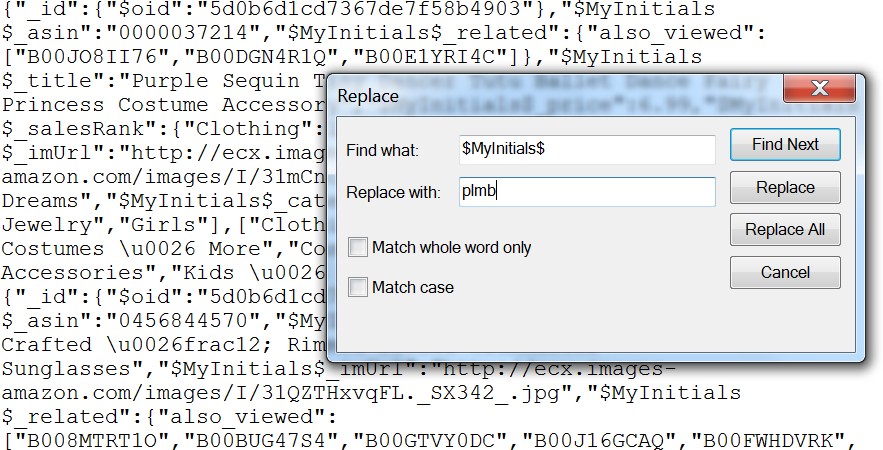
* products.zip
* reviews.zip

Unzip the files to extract two JSON files; they are called:

* $MyPNumber$\_products.json
* $MyPNumber$\_reviews.json

Rename the filenames by replacing the placeholder $MyPNumber$ (including the dollar signs) with your own student P Number. For example: P77342060\_products.json but type your P Number.

Open each JSON file in a text editor such as NotePad. Replace all occurrences of the placeholder $MyInitials$ (including the dollar signs) with your own set of initials. For example, plmb or jfk but replace with your own initials:



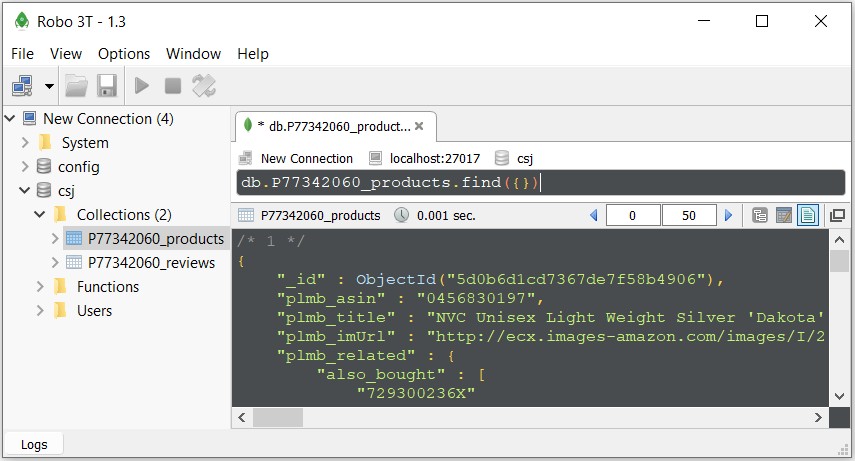
**WARNING**: The file containing reviews is particularly large and it may take the text editor several minutes to add your initials. Please be patient! You will only have to do this once.

Save and close the JSON files.

Then import the data into two new MongoDB collections so that the filenames (containing your own P Number) are used as the names of the collections. For example (but the filenames must contain your own P Number):

mongoimport /d csj /file P77342060\_products.json mongoimport /d csj /file P77342060\_reviews.json

This should create a personalised database called **csj** containing two collections: one for products and one for reviews. For example:



# NoSQL Data Samples

Here is a sample document from the **products** collection and a description of its fields (without your prefixed initials):

{

"\_id" : ObjectId("5d0b6d1cd7367de7f58b4a72"), "asin" : "B00005BHKS",

"related" : {

"also\_bought" : [ "B003JMEWNW", "B0042ZYLZO", "B000FCUS14", "B003IZ8SRQ", "B000WH47LE", "B003IYWE68", "B005V0H61A" ],

"also\_viewed" : [ "B0042ZWJNA", "B003IYWE68", "B003JMEWNW", "B009KAM6KS", "B004301YT4" ],

"bought\_together" : [ "B000FCUS14" ]

},

"title" : "Hogwart Student Hat Large", "price" : 19.95,

"salesRank" : {

"Toys & Games" : 127009

},

"imUrl" : ["http://ecx.images](http://ecx.images-/)- amazon.com/images/I/414N47PZSJL.\_SY300\_.jpg",

"brand" : "elope", "categories" : [

[ "Clothing, Shoes & Jewelry", "Novelty, Costumes & More", "Costumes & Accessories", "Hats & Caps",

"Kids & Baby" ]

],

"description" : "This hat is so plush it stands up on its own (or is it really magic?). Made of a velvety black material and emblazoned with the Hogwarts crest, this witch's hat, with an appropriately witchy curve to its peak and a nice stiff brim, would make any student at Hogwarts (or any other school) proud. Pair this hat with a black dress or cape and a broomstick and you've got a wicked Harry Potter-inspired costume. Nifty features include the adjustable headband and hidden inner pocket. --M. Sullivan"

}

* asin - ID of the product, e.g. [B00005BHKS](https://www.amazon.com/dp/B00005BHKS)
* categories - list of categories the product belongs to
* description – description of product
* title - name of the product
* brand – brand name of the product
* price - price in US dollars (at time of crawl)
* salesRank - sales rank information. The better the sales, the lower the rank.
* imUrl - url of the product image
* related - related products (also bought, also viewed, bought together, buy after viewing)

Here is a sample document from the **reviews** collection and a description of its fields (without your prefixed initials):

{

"\_id" : ObjectId("5d0b70f2d7367de7f5fa1d7e"), "reviewerID" : "A1RVTBD2QZ8TQ8",

"asin" : "B00005BHKS",

"reviewerName" : "A. Smith", "helpful" : [

1,

2

],

"reviewText" : "Amazing purchase and super fast shipping! This hat was so much better than I expected!!! I actually purchased it for myself and with the flex fit, it was absolutely perfect! I LOVE it!",

"overall" : 5.0,

"summary" : "LOVE, LOVE , LOVE!!",

"unixReviewTime" : 1387324800,

"reviewTime" : "12 18, 2013"

}

* reviewerID - ID of the reviewer, e.g. [A1RVTBD2QZ8TQ8](http://www.amazon.com/gp/cdp/member-reviews/A1RVTBD2QZ8TQ8)
* asin - ID of the product, e.g. [B00005BHKS](https://www.amazon.com/dp/B00005BHKS)
* reviewerName - name of the reviewer
* helpful - helpfulness rating of the review, e.g. 1 out of 2 found review helpful
* reviewText - text of the review
* overall - rating of the product
* summary - summary of the review
* unixReviewTime - time of the review (unix time in seconds)
* reviewTime - time of the review (raw)

# Query the collections

Write the following queries (Q1 to Q9) against the Amazon products and reviews collections that you personalised and loaded into MongoDB.

For each query, submit the MongoDB command in plain text, AND present it with its results as a screenshot showing the command and the documents returned.

For example:

//Q10. Find the details of the product identified as "B00006I55K".

db.P77342060\_products.find({plmb\_asin:"B00006I55K"})



However, you must use your own student P Number and initials.

If a lot of documents are returned, display them in the most appropriate Robo 3T mode, such as table mode**,** and show the first set of documents that are displayed. Ensure that what is displayed answers the requirements and demonstrates that the query is correct.

You must do this in order to gain all the marks available for a question.

Continued on next page …

Q1. Find the description and price of the product entitled “Kid's Astronaut Hat”. No other product details are required.

[8 marks]

Q2. Find all details of products that are priced at either $7.99 or $14.99.

[8 marks]

Q3. Count the number of products that have a price but do not have a description. Clearly show the number you receive as your answer.

[10 marks]

Q4. List the product IDs of products that were reviewed before the 15th June 2000. All calculations should be performed by MongoDB. Only list the product IDs and do not show any duplicates.

[10 marks]

Q5. List all of the reviewers who have written at least 3 reviews. Show the reviewer ID, reviewer name and the number of reviews for each one. List the names of reviewers in alphabetical order. [Hint: it is possible to solve using aggregate pipeline method].

[11 marks]

Q6. Count the number of related products that were also viewed with the product “Mens Timex”. Only give the number of related products as your answer. [Hint: it is possible to solve using aggregate pipeline method].

[6 marks]

Q7. List the product ID and title of all products in the “T-Shirts” category. No other product details are required. [Hint: it is possible to solve using aggregate pipeline method].

[7 marks]

Q8. Find the highest rating by the reviewer identified as “A2B3TVWTZ7609A” by using **map reduce** programming.

[8 marks]

Q9. Using both collections, find the title and description of products reviewed by the reviewer identified by “A2C65Q9IDNCSR”. No other product details are required.

[8 marks]

# Implement and Explain Index for the database

E1. Implement one index that would improve the querying of the database based on one or more of the queries (Q1-Q9). Identify the chosen query or queries and explain and justify your choice of index. Present, explain and compare execution plans to support your choice and summarise your findings.

[10 marks]

# Re-design the database

D1. Write code in MongoDB to automatically embed the details of reviews from the reviews collection with their corresponding product in the product collection.

Therefore, the products collection will contain both the product data and review data in a single collection of products.

This requires the following tasks:

s

1. Make new copies of the products and reviews collections and give them new names.
2. Embed all of the reviews into their corresponding product using the new collections.
3. In order to verify that the changes to the new products collection were successful, display the title and reviews of the product identified as “B00005JHK9”.

Tasks b and c must each be fully automated by writing code.

Note that task b) may take many seconds or minutes to execute depending on the machine. Therefore, in Robot3T, you are advised to select Change Shell Timeout … from the Options menu and set its value to a long time, say 600 seconds.

Note that there is no longer a need to reference the product ID inside each review now that it is part of the product.

[14 marks]

# Deliverables and Deadlines

You are required to upload your answers to the questions into Turnitin on Blackboard as one file. All questions and answers must be labelled with their capital letter and number (i.e. Q1 to Q9, E1 and D1). Add your P number to your filename. Your file must be in a readable format so the NoSQL code in plain text (Q1 to Q9, E1 and D1) can be copied and executed from it. In Turnitin, press both the Upload button and the Confirm button to submit your file and receive a receipt.

Deadline for both submissions: **Friday 26th March 2020, 13:00 or earlier**

Work will be considered late if uploaded to Blackboard after the deadline.

# Demonstration

You may be asked to demonstrate your work to your lab tutor. The aim of the demonstration is to give you the chance to show off all of the coding items that you implemented and to prove that they work. The tutor may ask you to demonstrate certain items and may ask you questions about how these items work. The demonstration will take place in the teaching weeks following the submission of your work. Your tutor will arrange a time for your demonstration which will normally take place in your timetabled lab. The tutor will prioritise those students whose submission does not show that the code works or their code does not appear to match their results. This will enable the student to gain marks which they would otherwise not obtain. If you do not attend the demonstration then you may receive a low mark or no mark at all.

# Closing comments

1. This is an individual assignment. Do not negotiate with others to clarify what is required. By uploading your work to Turnitin in Blackboard you will be declaring that the work you submit is your own and not plagiarised in any way.
2. Tutors are prepared to offer help with the initial part of the assignment where there is clear evidence that you have made a substantial attempt but have become stuck.
3. Tutors will offer help with later parts of the assignment, only where it is clear that you have a problem that no student could reasonably overcome. Typically, the solution will then be announced to all students. This is unlikely but not impossible.
4. Mark totals are shown alongside each question. Your tutor will need to able to execute your NoSQL code in MongoDB to check for correctness. Marks will be awarded for correctness and presentation. Marks will typically be deducted if information is erroneous, missing, irrelevant or difficult to ascertain. Marks can also be deducted if the answer is particularly inefficient. Therefore, partial marks are available if an answer is partially correct or partially presented. There is often more than one way to answer a question and so it is possible to gain full marks for an answer even if it is different from the markers' specimen set of answers. However, if these answers do not follow the examples and exercises taught on the module they

may be inferior in some way and so consequently marks may be deducted. Note that some questions hint at the most appropriate method to use.

1. Faculty policy for late work will apply. Any students not presenting this piece of work by the deadline will have their marks reduced unless formal procedures for the submission of late work are followed. If the work is handed in less than two weeks late it will be capped at 40% of the assignment mark. If the work is more than two weeks late then it will be capped at zero.

# References

Permission was granted to use the data for this coursework from the following source: <http://jmcauley.ucsd.edu/data/amazon/>on condition that the authors’ research papers are referenced. It is not necessary to read these papers in order to complete this coursework.

R. He, J. McAuley. "Modeling the visual evolution of fashion trends with one-class collaborative filtering". WWW, 2016.

J. McAuley, C. Targett, J. Shi, "A. van den Hengel. Image-based recommendations on styles and substitutes". SIGIR, 2015.