| | |
|---|---|
| **Weight:** | 40% of all marks for the unit. |
| **Due Date:** | Sem 2/2023 Week 8 - Monday, 11:55 pm |
| **Submission**: | (i) The assignment submission must be made through Moodle for this unit by Sem 2/2023 Week 8 - Monday, 11:55 pm<br><br>(ii) The submission of this assignment must be in the form of a single PDF file and a single ZIP file. No other forms will be accepted.<br><br>(iii) A Google doc file must be updated with updates made over time and match the submitted PDF folder |
| **Feedback:** | Feedback will be provided on student work within ten working days post submission via Moodle unless announced otherwise. |
| **Lateness:** | A penalty of 10% per day after the due date, including the weekends.<br>**Important:** Tasks submitted more than seven days after the due date will receive a mark of zero for that task. |
| **Learning outcomes:** | LO1. Describe various non-relational database systems, including NoSQL.<br>LO3. Design database systems using document-store techniques.<br>LO5. Implement and manipulate document-store database systems. |
| **Authorship:** | This assignment is an individual assignment, and the final submission must be identifiably your own work. Breaches of this requirement will result in an assignment not being accepted for assessment and may result in disciplinary action. |
| **Version Number** | **2.00** \| 15/08/2023 : Marking Rubric Adjusted to 40%<br>**3.00** \| 20/08/2023 : Sample Data for Task 1 Modified (refer to forum post)<br>**4.00** \| 20/08/2023 : B2.3 ~~seasonality ->~~ abundance,<br>**5.00** \| 22/08/2023 : Data Model Diagram added to Task 1 outputs. |

| | |
|---|---|
| | **6.00** \| 27/08/2023 : Updates made according to Ed forum post |

## Getting help and support:

What can you get help for?

- ***Consultations with the Teaching Team***
  Details can be found on **Moodle > Consultations**
- ***English language support/English Connect:***
  https://www.monash.edu/english-connect
- ***Learning skills support***:
  https://www.monash.edu/library/skills/contacts
- ***Counselling support:***
  https://www.monash.edu/health/counselling/appointments

## Extensions and Special Considerations:

If you are experiencing difficulties that you think will impact your ability to meet this deadline, you may apply for an assignment extension. You must apply **no later than two University working days after the due date** of this assignment.

The extension application can be found on ***Moodle > Assessments > How to Apply for an Extension***. Please allow **two business days** for your application to be processed.

Students should carefully read the Special Consideration website, especially the details about what formal documentation is required.

All special consideration requests should be made using the Special Consideration Application.

Please do not assume that submission of a Special Consideration application guarantees that it will be granted – you must receive an official confirmation that it has been granted.

## Plagiarism and Collusion:

Monash University is committed to upholding standards and academic integrity and honesty. Please take the time to view these links.

Academic Integrity Module
Student Academic Integrity Policy
Test your knowledge, collusion (FIT No Collusion Module)

**Generative AI tools cannot be used for any assessments in this unit.**

*In this unit, you must **not** use generative artificial intelligence (AI) to generate any materials or content in relation to your assessment.*

# FIT3176 Individual Assignment - Sem 2/2023 (Weight: 40%)
# Due date: Sem 2/2023 Week 8 - Monday, 11:55 pm

The assignment is divided into THREE main tasks:

**A. Data Modelling – Monash Travel Agency (MTA)**
**B. CRUD – Monash Nature Club (MNC)**
- **B.1. Modifying the Database**
- **B.2. Querying the Database**

**C. Charts and Drivers – Monash Nature Club (MNC)**

## A. Data Modelling – Monash Travel Agency (MTA) - 12%

**Case Study**

Monash Travel Agency (MTA), headquartered near Monash University Clayton Campus, is a new and popular travel initiative from Monash University. MTA offers a wide range of travel services, including transportations, hotel reservations, activity itineraries, and more for Monash Staff and Students required to travel domestic and international locations. However, with the recent increase in travel MTA is facing challenges in managing their operations efficiently due to the use of relational database management systems which is hindered by fixed data structures, manual processes, and long query processing times. As a result, to maintain their competitive advantage in the travel business and provide better customer experience, MTA has recognized the need for MongoDB, a modern NoSQL database solution.

The data required to be stored in MTA's MongoDB comes from their use case detailed below:

MTA allows booking to be made by customers and managed by agents. The agency has a fixed set of itineraries (e.g. travel to Malaysia Campus and Explore Perth on the way) which customers are able to book for particular dates throughout the year.

Itineraries contain details of accommodations, activities, transportations and travel dates. Accommodations are usually a fixed set of hotels for each location specific to the itinerary. Itineraries have a fixed set of activities to happen between a fixed number of days (e.g. Explore Malaysia for 7 days).

Transportations can be by car, flights, ships etc and each transportations has its own set of features and the details of the Transportations may change for each activity depending on availability.

Agents are also responsible for managing the finance side of the agency where they have to determine the monthly profit/loss of the agency by analysing all bookings and

forecasting which itineraries are profitable to keep and which to remove due unpopularity or additional expenses.

Given the above data requirements, create a Data Model Diagram for a MongoDB Database which can efficiently store the data required by Monash Travel Agency (MTA).

As part of the data model you are required to create at least one collection(s) / index(or indices) in the MongoDB database (named `<your_atuthcate>_MTA`) using the data provided in appendix A. Then, using MongoDB Compass provide screenshots of all created collection(s), index(/indices) and at least one document(s). Note each collection and index should be given relevant names (and are not required to include your authcate).

| | |
|---|---|
| **Task Outputs** | 1. Data Model Diagram<br>2. Screenshot of all created collection(s), document(s) and index(/indices) with each screenshot clearly displaying:<br>● connection string<br>● database name<br>● collection name<br>● total number of documents<br>● total number of indices<br>● expand embedded documents if any<br>3. Code for creating the collection(s), document(s) and index(/indices) |
| **Software to be use** | MongoDB Compass/Atlas |
| **Restrictions** | Screenshots should not be taken from MongoDB Shell |
| **Output files to use** | **<your authcate>_task_A.js**<br>code added for creating the collection(s), document(s) and index(/indices)<br><br>**<your authcate>_FIT3176_Assignment_1.gdoc**<br>Code and screenshot of collection(s), document(s) and index(/indices) |

| | |
|---|---|
| | **<your authcate>_Assignment_1.pdf**<br>Code and screenshot of collection(s), document(s) and index(/indices) |

**B. CRUD – Monash Nature Club (MNC) -14%**

Monash Nature Club (MNC) is recently studying the parks and different wildlife habitats across the United States in hopes of finding any links between the parks and wildlife of Australia.

The club has hired your team of Advanced Database Experts to use the following sample data files to help with the parks and wildlife analysis:

- wildlife.json
- parks.csv

**Note:** These data are raw data that does not follow any particular schema. For more information about the fields/columns in the data please refer to Appendix C.

For the analysis MAC has asked your team to perform the following tasks:

**B.1. Loading and Modifying the Database - 7%**

Create a MongoDB database named using the format: `<your_atuthcate>_MNC` Load the wildlife.json and parks.csv files using MongoDB Atlas/Compass into the newly created MongoDB database after creating one collection called `parks` and another collection called `wildlife`.

Using the newly created collections and one command for each subquestion (e.g. B1.1.) below perform the following modifications:

B.1.1. for each document in the wildlife collection, modify the common names by converting the comma separated values to array elements.

B.1.2. modify each document in the wildlife collection, to add a new field called `ts` with the time each document was created.
Note: you will need to find out first when each record in the document was created, not when they were inserted into the database e.g. try to find any field that indicates the creation date.

B.1.3. modify the parks collection to combine the date data (from parkEstDay, parkEstMonth and parkEstYear) to dateEst a date data type storing the combined year, month and data. Afterwards, remove the fields for year, month and day.
Note: For the date data other than day, month and year, the remaining components of the time can be taken as any value.

B.1.4. combine the data for parks into the wildlife collection and store the output in the wildlife collection, replacing the previous data in the wildlife collection. In the updated wildlife collection the park details should reside inside an array and should not contain a separate _id field.

B.1.5. Modify the database to ensure that all new data contains no new values (allowing only values already in the database) for recordStatus, occurrence, nativeness and abundance. Demonstrate the new modification is working by incorrectly adding new data to the wildlife collection.
Note: you will need to first check what are the existing categories/values for recordStatus, occurrence, nativeness and abundance, and then ensue if new data is entered it only contains values from the existing categories

**B.2. Querying the Database - 7%**
For each question, use one aggregation query to answer using the resulting collection from B.1. Unless stated otherwise the queries must not add/remove/modify fields in the database.
**Note:** Marks for this section depend on the query efficiency e.g. the processing speed, the storage, the number of queries used etc.
Therefore, using too many queries/indices to answer a section or using temporary variables (such as const, var etc), collections, for each loop may incur mark penalties.

For each question in this section you are expected to:
use your own judgement when it comes to query efficiency
provide justification for each index (if any are created)
include query execution plans from MongoDB compass

B.2.1. What was the total number of parks established each year? The output should be ordered by highest to lowest year and display the distinct year and number of parks established.

B.2.2. What was the average number of Bird category species in each park? The output should display the distinct park name and the average number of Birds.

B.2.3. List all parks within 25 km and 400km of the park with the most Uncommon ~~seasonality~~ abundance species. The list should not show duplicate park names. (if required for this query only the collection can be modified).

B.2.4. Display the species with the most common names that belong to the same category. The output should display the species id, the category and number of common names. ~~distinct park name and the average number of Birds~~.

B.2.5. Display the total number of distinct categories, orders and families in each park for all species with common names starting with the letter “a”. The output should display the distinct category, order and family names and the count rounded to 2 decimal places.

| | |
|---|---|
| **Task Inputs** | The data for this task is contained in the following files:<br>(i) wildlife.json<br>(ii) parks.csv |
| **Task Outputs** | Screenshots for each question<br>B.1.From MongoDB Compass/Atlas at least one screenshot of the modified document before modifications were made and at least one screenshot after modifications were made, with each screenshot clearly displaying:<br>• connection string<br>• database name<br>• collection name<br>• total number of documents<br>• total number of indices<br>• expand embedded documents if any<br>B.1. Screenshot of successful query execution message from MongoDBShell<br><br>B.2.Screenshot from MongoDBShell for the code executed |

| | |
|---|---|
| | Code for each question<br>**Properly commented code** for all subquestions in B.1. and B.2. in one file |
| **Software to be use** | MongoDB Compass/Atlas for data load and execution plans.<br><br>MongoDB Shell for executing the code. |
| **Output files to use** | **<your authcate>_task_B.js**<br>Code added for each sub question<br><br>**<your authcate>_FIT3176_Assignment_1.gdoc**<br>Code and screenshot for each sub question<br><br>**<your authcate>_Assignment_1.pdf**<br>Code and screenshot for each sub question |

**C. Charts and Drivers - 14%**

**C.1. MongoDB Charts - 6%**

Create a MongoDB Charts dashboard named <your authcate>_Assignment1

For each question in task B.2. Querying the Database, create a mongoDB charts visualisation to visualise the results. You may use your own judgement with appropriate justifications provided in the report when selecting the type of chart to clearly display your findings.

**C.2. MongoDB Driver - 4%**

Create a runnable python script using the pymongo driver to display the result of each question in task B.2 and save it named <your authcate>_task_C2.py (must not be a Jupyter notebook file)

Your script should:
be executable using the terminal and command prompt,
contain adequate comments

output the query description and query results on the terminal and command prompt when run.

| | |
|---|---|
| **Task Inputs** | The data for this task is contained in the following files:<br>(i) wildlife.json<br>(ii) parks.csv |
| **Task Outputs** | Screenshots for each question<br>Screenshots for C.1 from MongoDB Atlas clearly displaying:<br>• dashboard name<br>• entire dashboard<br>• edit chart section of each individual chart<br>Code for each question<br>All required **properly commented python code** to implement C.2. |
| **Software to be use** | MongoDB Atlas for Charts Screenshots.<br><br>A software that can run scripts such as Python script commands (e.g. VS Code etc.). |
| **Output files to use** | **<your authcate>_task_C.py**<br>Code added for each sub question<br><br>**<your authcate>_FIT3176_Assignment_1.gdoc**<br>Code and screenshot for each sub question<br><br>**<your authcate>_Assignment_1.pdf**<br>Code and screenshot for each sub question |

### C.3. Attention to Detail - 4%

Less than 4 updates on different days in google doc -4%

OR Incorrect numbers of files/Incorrect file types -4%

OR Inconsistent data in screenshots and/or code between files -4%

## D. General Information and Submission Checklist

- *Submission method*: Submission is online through Moodle.
- *Penalty for late submission*: 10% deduction for each day (including weekends).
- *Assignment Cover Sheet*: You will need to sign and attach the assignment cover sheet as part of the pdf file.
- Please carefully read the requirements for EACH section, especially the Task Outputs and Restrictions.
- You will be required to regularly upload your work to Google Docs (for tracking the history of your work)

**Generative AI tools cannot be used for any assessments in this unit.**

*In this unit, you must **not** use generative artificial intelligence (AI) to generate any materials or content in relation to your assessment.*

D.1. One **combined pdf file** containing all tasks mentioned above and named **<your authcate>_Assignment_1.pdf**

The pdf file should contain:

i. Cover page

ii. A signed cover sheet

iii. A report that combines all the tasks (including explanations, code and screenshots)

***Note:** It is expected that this PDF document is quite large due to incorporating the above components. Please note that the **report readability also contributes to your assignment mark.***

D.2. Two executable javascript files for Task A and B

**Note:** All of the above files must be runnable in MongoDBShell.
You should also clearly indicate each task using comments and follow appropriate naming conventions e.g. as specified in the MongoDB documentations.

**Penalties may apply for any inconsistencies between the screenshots in the PDF report and the outputs of the script files.**

D.3. Driver script files (i.e. .py file must not be a Jupyter notebook file) for Task C.2.
**NOTE: All submitted code must be properly formatted and commented.**

D.4. The code files from D.2. and D.3. must be zipped into a folder named **<your authcate>_FIT3176_A1Code.zip**.
**NOTE: All submitted code must be properly formatted and commented.**

D.5. Upload both the **PDF file** and **ZIP file** to Moodle.

The submission of this assignment must be in the form of **a single PDF file (<your authcate>_FIT3176_Assignment_1.pdf), a single ZIP file (<your authcate>_FIT3176_A1Code.zip), and an updated over time google doc file (<your authcate>_FIT3176_Assignment_1.gdoc).**
No other forms will be accepted.

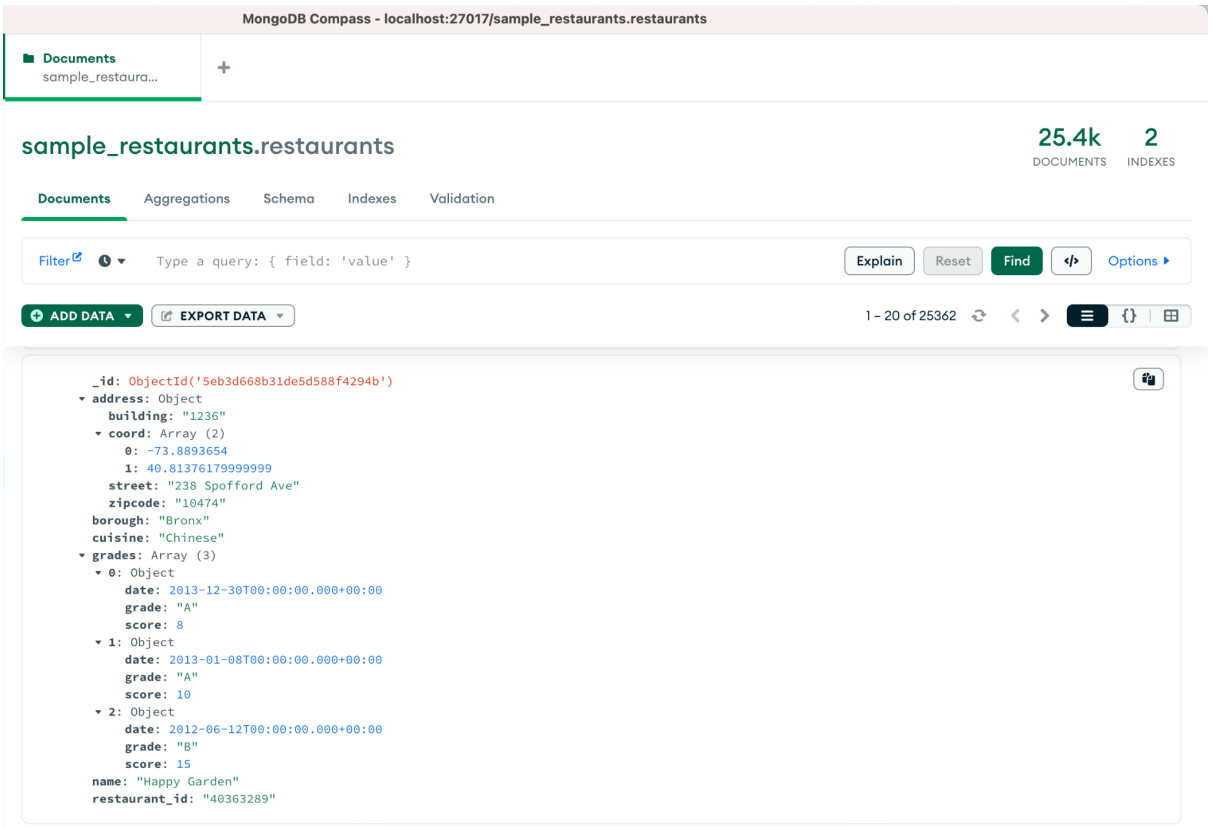
## Appendix A. MTA's Previous Database

| Table | Fields | Example |
|---|---|---|
| Customer | Customer id | C_0001 |
| | Customer name | John Smith |
| | Customer contact | 0123456789 |
| | Customer address | 4 Clayton Ave, Clayton, 3800, VIC |
| Agent | Agent id | A_0001 |
| | Agent name | Harry Brown |
| | Agent contact | 0123456788 |
| | Agent office address | 4 Wellington Ave, Clayton, 3800, VIC |
| Booking | Booking id | B_0001 |
| | Customer id | C_0001 |
| | Booking date | 2023-09-12 |
| | Booking agent | Harry Brown |
| | Itinerary id | I_0001 |
| | Booking cost ($ discounted) | $1200 |
| Itinerary | Itinerary id | I_0001 |
| | Itinerary description | Explore Malaysia and drop by Perth on the way |
| | Itinerary start date | 2024-09-12 |
| | Itinerary end date | 2024-10-05 |
| | Itinerary total cost | $1250 |
| Activity | Activity id | A_0001 |
| | Activity description | Explore Perth |
| | Activity no of days | 3 |
| Activity | Activity id | A_0002 |
| | Activity description | Explore Malaysia |
| | Activity no of days | 7 |
| Itinerary_Activity | Activity id | A_0001 |
| | Itinerary id | I_0001 |
| | Activity start date | 2024-09-13 |
| | Activity end date | 2024-09-16 |
| Itinerary_Activity | Activity id | A_0002 |
| | Itinerary id | I_0001 |

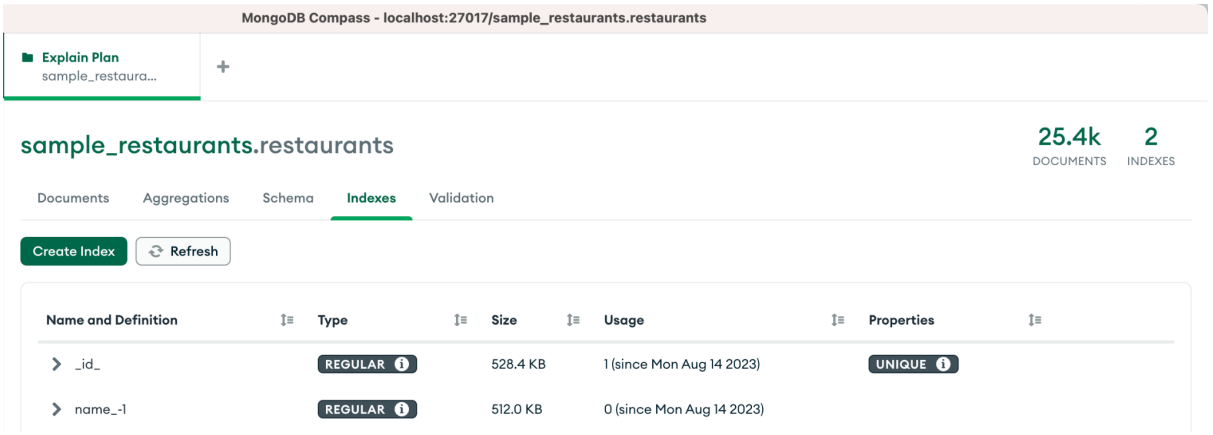
| | | |
|---|---|---|
| | Activity start date | 2024-09-17 |
| | Activity end date | 2024-10-04 |
| Transportation_car | Transportation id | T_0001 |
| | Activity id | A_0002 |
| | Transportation type | Hired Car |
| | Transportation booking start date | 2024-09-12 |
| | Transportation booking end date | 2024-09-16 |
| | Transportation usage | Travel from Melbourne to Perth and then travel around Perth |
| Transportation_flight | Transportation id | T_0002 |
| | Activity id | A_0001 |
| | Transportation type | Commercial Flight |
| | Flight departure date | 2024-09-16 |
| | Flight departure airport | Perth Airport |
| | Flight arrival date | 2024-09-17 |
| | Flight arrival airport | Malaysia Airport |
| | Flight Airline career | Malaysian Airlines |
| Accommodation | Accommodation id | ACC_001 |
| | Accommodation name | Hilton Perth |
| | Accommodation address | 27 Perth road, Perth 1111, WA |
| | Accommodation check-in date | 2024-09-13 |
| | Accommodation check-out date | 2024-09-16 |

# Appendix B.

Screenshot format for collection(s), documents and embedded documents (if any):




Screenshot format for index(/indices):



# Appendix C.

## Dataset details

wildlife.json:

| Field | Description |
|---|---|
| speciesID | unique id for each species |
| parkName | name of the park where the species can be found |
| category | category of the species |
| order | order of the species |
| family | family of the species |
| scientificName | scientific name the species (genus, species, subspecies) |
| commonName | one or more common names of the species it is commonly known by |
| recordStatus | the park record status of the species |
| occurrence | shape of the observed UFO sighting |
| nativeness | species native or foreign to park |
| abundance | presence and visibility of species in park |

parks.csv:

| Field | Description |
|---|---|
| parkName | name of the park |
| stateCode | code of the state or states where the park is located |
| areaInAcres | area or size of the park in acres |
| latitude | latitude of the park centre |
| longitude | longitude of the park centre |
| parkEstDay | day of the month the park was established |
| parkEstMonth | month of the year the park was established |
| parkEstYear | year the park was established |

# References

National Park Service. (2017, January 20). Biodiversity in National Parks [Dataset]. Kaggle. https://www.kaggle.com/nationalparkservice/park-biodiversity?select=species.csv