Kirirom Institute of Technology

FULL STACK DEVELOPMENT BOOTCAMP

Module 2 (Database) - Mongodb

Instructions:

- Each exercise carries 10 points.
- Each exercise contains Description of the problem, requirement, hint and expected output.
- Read the instruction clearly before practicing the exercises.
- The answer sheet must be named after the naming convention mentioned in each exercise.
- Each exercise done by student is expected to meet the learning outcome mentioned in each exercise.
- All the exercises must be completed on time.
- You may discuss about the problem conceptually with the trainer or fellow students or from internet **however getting direct answers or copying code from others or from internet is strictly prohibited.**
- All the exercise files must be submitted to google classroom.
- All the exercises will be evaluated and points will be updated by the end of each week.
- Based on the points obtained by each student the overall ranking will be maintained.
- **All the documents (including this document) used in this bootcamp are sole properties of KIT and for internal purposes only. Must not be shared to anyone outside of this bootcamp.**

EXERCISE_11:

Description:

- Update the existing records with the "**name"** property with the value **"Phalin"** in the **"records"** collection of the **"employee"** database as a single operation using command prompt.

  Assume below is the current set of values of the row:

  {

  id: 1250
  name: "Phalin"
  phone: 132345
  current_date: <Current Date>

  }

  The value to be updated mentioned below.

  {

  id: 19990
  name: "Chan Phalin"

  }

- Note. If the record/records not found then create a new record. The extra columns (phone and current_date) apart from the new set of values must be removed from the document.
- Display the updated records in a proper format after update.

Requirements:

- Screenshots of the steps and results must be added to the word file **11_Mod_02_update_records01.docx** and to be submitted to the google classroom.

Hint:

- update()

Expected Outcome:

To learn how to update record/records from the collection in a single operation (Affecting the other properties of that record).

EXERCISE_12:

Description:

- Update the existing records with the "**name"** property with the value **"Vichea"** in the **"records"** collection of the **"employee"** database as a single operation using command prompt.

id: 134
name: "Vichea"
phone: 12345
current_date: <Current Date>

The value to be updated mentioned below.

id: 14
name: "Ratanak Vichea"

- Note. If the record/records not found then create a new record. The extra columns (phone and current_date) apart from the new set of values **must still remain the same in the existing record if the values are not passed in the update()**.
- Display the updated records in a proper format after update.

Requirements:

- Screenshots of the steps and results must be added to the word file **12_Mod_02_update_records02.docx** and to be submitted to the google classroom.

Hint:

- update()
- $set

Expected Outcome:

To learn how to update record/records from the collection in a single operation without affecting the other properties of the record.

EXERCISE_13:

Description:

- Assume there are three records already available in the collection **"records"** of the **"employee"** database as mentioned below.

  id: 134
  name: "Vichea"
  phone: 12345
  salary: 10000
  current_date: <Current Date>

  id: 1250
  name: "Phalin"
  phone: 132345
  salary: 11000
  current_date: <Current Date>

  id: 126
  name: "Vichea"
  current_date: <Current Date>
  address: { street: "West street", city : "Phnom Penh" }
  salary: 20000
  favourites: ["Music", "Reading books", "Singing"]

- Update the existing records and increment **"salary" by 1000** for the records whose **"name"** is **"Phalin"**.

  id: 134
  name: "Vichea"
  phone: 12345
  salary: 10000
  current_date: <Current Date>

  id: 1250
  name: "Phalin"
  phone: 132345
  salary: 12000

current_date: <Current Date>

id: 126
name: "Vichea"
current_date: <Current Date>
address: { street: "West street", city : "Phnom Penh" }
salary: 20000
favourites: ["Music", "Reading books", "Singing"]

- Display all the records in a proper format after update.

Requirements:

- Screenshots of the steps and results must be added to the word file **13_Mod_02_update_records_increment_03.docx** and to be submitted to the google classroom.

Hint:

- update()
- $inc

Expected Outcome:

To learn how to read a particular record/records from the collection in a single operation (Similar to using where conditions in Relational Databases)

EXERCISE_14:

Description:

- Update the name of the **"phone"** property in the **"records"** collection of the **"employee"** database into **"mobile"**. Let's assume the current record in the collection looks like below.

  id: 134
  name: "Vichea"
  phone: 12345
  current_date: <Current Date>

  After updating the property should look like below.

  id: 134
  name: "Vichea"
  mobile: 12345
  current_date: <Current Date>

- Display the updated records in a proper format after update to check if the property name is changed.

Requirements:

- Screenshots of the steps and results must be added to the word file **14_Mod_02_rename_property.docx** and to be submitted to the google classroom.

Hint:

- update()
- $rename

Expected Outcome:

To learn how to update record/records from the collection and to change the property names.

EXERCISE_15:

Description:

- Remove an existing record/records with the "**name"** property with the value **"Vichea"** in the **"records"** collection of the **"employee"** database as a single operation using command prompt.
- Display the latest records in a proper format after deleting.

Requirements:

- Screenshots of the steps and results must be added to the word file **15_Mod_02_delete_records.docx** and to be submitted to the google classroom.

Hint:

- remove()

Expected Outcome:

To learn how to delete record/records from the collection based on a condition.

EXERCISE_16:

Description:

- Create the two records with the below properties and values into the **"records"** collection of the **"employee"** database.

id: 134
name: "Vichea"
phone: 12345
salary: 10000
current_date: <Current Date>

id: 1250
name: "Phalin"
phone: 132345
salary: 11000
current_date: <Current Date>

- After creating the above records now add **subdocument** named **"reviews"** to each record mentioned above. The **"reviews"** sub document must have three properties named **reviewer_name, message_body and current_date.** The review for each record is mentioned below.

Reviews for the record of **"Vichea":**

reviews: [{

reviewer_name: "Arun",

message_body: "He is a good Employee",

current_date: <Date>

},

{

reviewer_name: "Reak",

message_body: "His performance is good",

current_date: <Date>

}]

Reviews for the record of **"Phalin":**

reviews: [{

reviewer_name: "Vandate",

message_body: "His coding skills are very good",

current_date: <Date>

}]

Requirements:

- Screenshots of the steps and results must be added to the word file **16_Mod_02_adding_sub_documents.docx** and to be submitted to the google classroom.

Hint:

- Learn to add sub documents

Expected Outcome:

To learn how to delete record/records from the collection based on a condition.

EXERCISE_17:

Description:

- This exercise is the continuation of the previous exercise. After the **"reviews"** sub documents are added to the records. Now find and display the entire record where **"Vandate"** has reviewed.

Requirements:

- Screenshots of the steps and results must be added to the word file **17_Mod_02_find_review_from_sub_document.docx** and to be submitted to the google classroom.

Hint:

- find()
- Use **$elemMatch**

Expected Outcome:

To learn how to find a particular property from the sub document using elemMatch.

EXERCISE_18:

Description:

- Insert the below records to the **"records"** collection of the **"employee"** database.

  id: 134
  name: "Vichea"
  phone: 12345
  salary: 10000
  current_date: <Current Date>

  id: 1250
  name: "Phalin"
  phone: 132345
  salary: 11000
  current_date: <Current Date>

  id: 1505
  name: "Veasna"
  phone: 100307
  current_date: <Current Date>

- Now add indexing to the **"name"** property of the **"records"** collection. Perform a **text search** and find the records with the name that starts with "V" and ends with "a".

Requirements:

- Screenshots of the steps and results must be added to the word file **18_Mod_02_add_index_perform_text_search.docx** and to be submitted to the google classroom.

Hint:

- find()
- Use **$search**

Expected Outcome:

To learn how to find add indexing for a particular document and text search using **$search**.

EXERCISE_19:

Description:

- Insert the below records to the **"records"** collection of the **"employee"** database.

  id: 134
  name: "Vichea"
  phone: 12345
  salary: 5000
  current_date: <Current Date>

  id: 1250
  name: "Phalin"
  phone: 132345
  salary: 11000
  current_date: <Current Date>

  id: 1505
  name: "Veasna"
  phone: 10307
  current_date: <Current Date>

- Find and display the records where salary is greater than 5000
- Find and display the records where salary is less than or equal to 9000
- Find and display the records where salary is between 8000 and 11000

Requirements:

- Screenshots of the steps and results must be added to the word file **19_Mod_02_comparison.docx** and to be submitted to the google classroom.

Hint:

- $gt, $lt, $gte, $lte
- find()

Expected Outcome:

To learn how to find select records between a range.

EXERCISE_20:

Description:

- In this exercise you need to create a cloud database using Mongodb Atlas and perform the database operations from your command prompt.
- Visit: https://cloud.mongodb.com/v2/5f6b291937f71250ada0b378#clusters/pathSelector and login with your KIT email id.
- Choose the Sand box M0 plan which is free forever.
- Create cluster and name it as <your_name>_cloud_db.
- Under database Access option -> Create user with the name :<Your name> and give read and write permission.
- Under Network Access option -> Provide IPWhitelist option the permission to access from anywhere.
- Under Clusters option -> Choose connect -> Choose connect with the Mongo Shell and copy the connection string paste it in the mongo shell and enter.
- After connecting your mongo shell with the cloud. Now create a database or use any of the existing databases and create a collection named **"test"** from your Mongo shell (Command Prompt)**.**
- Insert a record into the collection from command prompt.

ALERT

Make sure that you follow all instructions properly. Be aware that each character is important. Upload your solutions for all the exercises on the google classroom created and shared with you.