



Ejercitación Document-Database con MongoDB

- 1) Instale la Base de Datos Documental llamada [MongoDB](#) en su versión 4.2.X..
 - a) (opcional) usando docker: docker pull mongo:4.2.3
- 2) Conectándose a la Base de Datos mediante la consola o una interfaz gráfica, ingrese los comandos para:
 - a) Liste las bases de datos existentes
 - b) Crear e ingresar a la base de datos: bd2_{SU Legajo}
 - c) Cree una colección llamada “equipo_docente” que contenga los siguientes datos:

- nombre: guillermo cherencio, cargo: titular, dedicación: semi-exclusiva, jefe_area: true, salario: 0

- nombre: juan carlo romero, cargo: ayudante de primera, dedicación: semi-exclusiva, horas: 100

- nombre: jose racker, cargo: ayudante de primera, dedicacion: simple, horas: 20, salario: 90

- nombre: carlos rodriguez, cargo: titular, dedicación: simple, horas: 90, salario: 90, estado: licencia

- nombre: walter panessi, cargo: titular, dedicacion: simple

- nombre: lucas mufato, cargo: ayudante de segunda, recibido: False

- d) Cree una query que devuelva un miembro de la colección
- e) Cree una query que devuelva todos los miembros de la colección
- f) Cree una query que devuelva todos los miembros con cargo “ayudante de primera”
- g) Cree una query que devuelva todos los miembros con cargo “ayudante de primera” y con dedicacion “simple”
- h) Cree una query que devuelva todos los miembros con cargo “titular” pero que no tengan estado = “licencia”
- i) Cree una query que devuelva todos los miembros con “horas” mayor a 50 o salario menor a 50
- j) MODIFIQUE el atributo “horas” para el nombre “juan carlos romero”, el nuevo valor sera 900
- k) MODIFIQUE el atributo “nombre” para que sea “nombre_apellido” en TODOS los registros
- l) ELIMINE el registro con nombre_apellido = “walter panessi”



UNIVERSIDAD NACIONAL DE LUJÁN
Departamento de Ciencias Básicas, División Sistemas
Licenciatura en Sistemas de Información (RES.HCS 009/12)
11078 Base de Datos II

3) Usando colecciones nuevas realizar los siguientes pasos:

a) Insertar los siguientes facturas (las fechas se deben ingresar como tipo de dato date):

```
{"id_fac":"1", "estado":"iniciada", "importe":0, "fecha":"2020/2/20"},
{"id_fac":"2", "estado":"finalizada", "importe":45, "fecha":"2020/2/21",
"detalles":[
  {"id_prod":1, "cantidad":1, "precio":25},
  {"id_prod":2, "cantidad":1, "precio":20}
]},
{"id_fac":"3", "estado":"anulada", "importe":270, "fecha":"2020/2/21", "detalles":[
  {"id_prod":3, "cantidad":10, "precio":25},
  {"id_prod":5, "cantidad":1, "precio":20}
]},
{"id_fac":"4", "estado":"finalizada", "importe":75, "fecha":"2020/2/22",
"detalles":[
  {"id_prod":1, "cantidad":1, "precio":25},
  {"id_prod":6, "cantidad":1, "precio":50}
]},
{"id_fac":"5", "estado":"iniciada", "importe":87, "fecha":"2020/2/22", "detalles":[
  {"id_prod":2, "cantidad":2, "precio":18},
  {"id_prod":4, "cantidad":2, "precio":23},
  {"id_prod":5, "cantidad":1, "precio":15}
]},
{"id_fac":"6", "estado":"iniciada", "importe":0, "fecha":"2020/2/23"}
```

- b) Crear query que retorne las facturas con estado iniciadas y fecha anterior a 2020/2/22
- c) Crear query que borre las facturas con estado anuladas y fecha anterior o igual a la actual
- d) Crear query que retorne las facturas que vendieron el producto con id 2
- e) Crear query que retorne las facturas que vendieron más de un producto
- f) Crear query que retorne las facturas que vendieron más de 3 unidades de un producto
- g) Crear query que retorne las cuyo primer producto vendido tiene el ID 3
- h) Crear query que retorne las facturas que no tiene detalles
- i) Crear una query que retorne el ID_FAC de las facturas que no tiene detalles



UNIVERSIDAD NACIONAL DE LUJÁN
Departamento de Ciencias Básicas, División Sistemas
Licenciatura en Sistemas de Información (RES.HCS 009/12)
11078 Base de Datos II

- j) Crear una query que retorne el ID_FAC y la fecha de las facturas que no tiene detalles, ordenando por fecha de más nueva a mas vieja
- 4) Expandir la base de datos (lea los primeros 3 puntos antes de realizar cambios) y realice las siguientes consultas:
- a) Inserte los productos:
- ```
{"id_prod":1, "descr": "regla 25cm", "stock":50, "precio":25, "precio_base":15},
{"id_prod":2, "descr": "lapis bic", "stock":200, "precio":20, "precio_base":7.5},
{"id_prod":3, "descr": "plasticola", "stock":50, "precio":100, "precio_base":50},
{"id_prod":4, "descr": "lapicera bic", "stock":200, "precio":25, "precio_base":15},
{"id_prod":5, "descr": "goma bic", "stock":100, "precio":15, "precio_base":5},
{"id_prod":6, "descr": "regla 35cm", "stock":20, "precio":50, "precio_base":35}
```
- b) Inserte los clientes:
- ```
{"id_cli":1, "descr": "consumidor final", "tipo_factura":"B"},
{"id_cli":2, "descr": "cherencio", "tipo_factura":"B"},
{"id_cli":3, "descr": "Escolares S.A", "tipo_factura":"A", "cuit":"30-60495864-0"},
{"id_cli":4, "descr": "SSA Utiles", "tipo_factura":"A", "cuit":"34-50004533-9"}
```
- c) Haga las modificaciones necesarias para que cada detalle de la factura esté relacionado con el producto y que cada cliente tenga una factura relacionada y cada factura un cliente. Usó un modelo normalizado o embebido? Por que? Que modificaciones tuvo que realizar?
- d) Muestre por cada cliente sus facturas
Ej: { id_cli:1, facturas: [{id_factura:1 ,.....},], {id_cli: 2}
- e) Muestre por cada factura los datos del cliente
Ej: { id_fac: 1, cliente: [{id_cli:1,},],}, {id_fac: 2}
- f) Muestre por cada cliente SOLO su id y el ID de sus facturas
Ej: { id_cli:1, facturas: [{id_factura:1 },] }, {id_cli: 2}
- g) Muestre las facturas del cliente cherencio
- h) Muestre la descripción del cliente con la factura 3
- 5) Diseñe el esquema de una aplicación de preguntas y respuestas multiple choice, cada pregunta podrá tener entre 3 a 5 respuesta, donde solo una es la correcta. Las preguntas están agrupadas en niveles dependiendo de su dificultad.
Un usuario pedirá una pregunta, tiene X tiempo para responder y al responder correctamente 2 preguntas pasa al nivel siguiente, al equivocarse empieza de cero.
Comparelo con un esquema relacional? Qué ventajas encuentra? Que problemas?



UNIVERSIDAD NACIONAL DE LUJÁN
Departamento de Ciencias Básicas, División Sistemas
Licenciatura en Sistemas de Información (RES.HCS 009/12)
11078 Base de Datos II

6) Dado el siguiente JSON escriba el esquema de validación para la colección “productos”

```
{
  "Sku": $int,
  "Nombre": $string,
  "Descripcion": $string,
  "Fecha_creacion": $date,
  "Imagenes": [$String],
  "Precio": {
    "Moneda": $string,
    "Cantidad": $double
  }
}
```

Todos los campos son obligatorios. Sku es un número entre 1.000 y 100.000. Imágenes debe contener al menos un string. Moneda debe contener el valor “US”, “EU” o “AR”

7) Cree una aplicación que interactúe con la BD diseñada en el punto 5 que permita jugar y gestionar el juego.
Enfóquese en aquello más importante para tener un MVP (Minimo producto viable) como: ABM de preguntas, jugar 3 preguntas de nivel 1 y pasar al siguiente nivel, perder y volver al principio. (puede hardcodear lo necesario para lograr el objetivo)