

PRUEBA FINAL PRESENCIAL

Unidad Curricular
Base de Datos NoSQL

2019-03

UTEC

Tabla de contenido

1.- Ejercicios MongoDB	3
1.1.- Crear las sentencias a ejecutarse en MongoDB.	3
1.2.- Generar una sentencia que permita recuperar los documentos cuyo Autor sea Edgar Allan Poe.	5
1.3.- Generar una sentencia que permita recuperar los documentos que tengan Precio mayor a 35 y la EdadComprador mayor a 40.	6
1.4.- Unificar los valores de Género, para que queden normalizados (que los valores finales sean Hombre y Mujer).	6
1.5.- Agregar la clave Mes, donde se cargue el mes que se realizó la compra (filtrar para Febrero, Marzo y Abril)	7
1.6.- Agregar la clave Clasico, donde se le cargue el valor SI cuando el Autor es Dumas o Dickens.	8
1.7.- Agregar la clave NivelEducativo, donde se le cargue el valor Alto si tiene formación Universitaria (completa o incompleta).	9
1.8.- Eliminar los documentos que no tengan clave FechaCompra	10
2.- Parte Java:.....	11
2.1.- Crear la colección VentasJava que registre los documentos de la planilla Datos.xlsx. Los datos deben cargarse como en la planilla, respetando las mayúsculas y minúsculas y solo cargándoles la clave cuando existen valor. La clave Formación es de tipo arra.	11
2.2.- Generar una sentencia que permita recuperar los documentos cuyo Autor sea Edgar Allan Poe.	11
2.3.- Generar una sentencia que permita recuperar los documentos que tengan Precio mayor a 35 y la EdadComprador mayor a 40.	11
2.4.- Unificar los valores de Género, para que queden normalizados (que los valores finales sean Hombre y Mujer).	11
2.5.- Agregar la clave Mes, donde se cargue el mes que se realizó la compra (filtrar para Febrero, Marzo y Abril).	11
2.6.- Agregar la clave Clasico, donde se le cargue el valor SI cuando el Autor es Dumas o Dickens.	11
2.7.- Agregar la clave NivelEducativo, donde se le cargue el valor Alto si tiene formación Universitaria (completa o incompleta).	11
2.8.- Eliminar los documentos que no tengan clave FechaCompra	12
3.- Parte teórica.	13

Dados los siguientes documentos:

CódigoInventario	Autor	Título	Precio	FechaCompra	EdadComprador	Género	Ciudad	Formación
123321	Alejandro Dumas	Los tres mosqueteros	10	01/02/2019	80		Lima	Primaria Completo, Secundaria Completo
135426	Arthur Conan Doyle	Las aventuras de Sherlock Holmes	11	02/02/2019	45	F		Primaria Completo, Secundaria Completo, Universidad incompleto
135426	Arthur Conan Doyle	Las aventuras de Sherlock Holmes	11	03/03/2019		Mujer	París	
235678	Charles Dickens	Cuento de Navidad	25		34			
987654	Charles Dickens	Historia de dos ciudades	30		57	M	París	
986532	Charles Dickens	Oliver Twist	12	02/04/2019		M		Primaria Completo, Secundaria Completo, Universidad completo
876521	Edgar allan Poe	El gato negro	40		27			
124567	edgar allan poe	Los crímenes de la Rue Morge	45	03/04/2019		Hombre	Buenos Aires	Primaria Completo, Secundaria Incompleto
234590	Fedor Dostoiewski	Crímen y castigo	40		59	Hombre		Primaria Completo, Secundaria Incompleto
135426	Arthur Conan Doyle	Las aventuras de Sherlock Holmes	29			F	Montevideo	

1.- Ejercicios MongoDB

1.1.- Crear las sentencias a ejecutarse en MongoDB.

Crear la colección Ventas que registre los documentos de la planilla Datos.xlsx. Los datos deben cargarse como en la planilla, respetando las mayúsculas y minúsculas y solo cargándoles la clave cuando existen valor. La clave Formación es de tipo array. Y la colección de documentos se debe llamar "**Ventas**".

```
use Prueba-Presencial-2019-04
db.createCollection("Ventas")
```

```
db.Ventas.insertMany([
  {
    "Código Inventario": "123321",
    "Autor": "Alejandro Dumas",
    "Título": "Los tres mosqueteros",
    "Precio": 10,
    "FechaCompra": new Date("2019-02-01"),
    "EdadComprador": 80,
    "Ciudad": "Lima",
    "Formación": ["Primaria Completo", "Secundaria Completo"]
  },
  {
    "Código Inventario": "135426",
    "Autor": "Arthur Conan Doyle",
    "Título": "Las aventuras de Sherlock Holmes",
    "Precio": 11,
    "FechaCompra": new Date("2019-02-02"),
    "EdadComprador": 45,
    "Género": "F",
    "Formación": ["Primaria Completo", "Secundaria Completo",
"Universidad incompleto"]
  },
  {
    "Código Inventario": "135426",
    "Autor": "Arthur Conan Doyle",
    "Título": "Las aventuras de Sherlock Holmes",
    "Precio": 11,
    "FechaCompra": new Date("2019-03-03"),
    "Género": "Mujer",
    "Ciudad": "París"
```

```

    },
    {
        "Código Inventario": "235678",
        "Autor": "Charles Dickens",
        "Título": "Cuento de Navidad",
        "Precio": 25,
        "EdadComprador": 34
    },
    {
        "Código Inventario": "987654",
        "Autor": "Charles Dickens",
        "Título": "Historia de dos ciudades",
        "Precio": 30,
        "EdadComprador": 57,
        "Género": "M",
        "Ciudad": "París"
    },
    {
        "Código Inventario": "986532",
        "Autor": "Charles Dickens",
        "Título": "Oliver Twist",
        "Precio": 12,
        "FechaCompra": new Date("2019-04-02"),
        "Género": "M",
        "Formación": ["Primaria Completo", "Secundaria Completo",
"Universidad completo"]
    },
    {
        "Código Inventario": "876521",
        "Autor": "Edgar allan Poe",
        "Título": "El gato negro",
        "Precio": 40,
        "EdadComprador": 27
    },
    {
        "Código Inventario": "124567",
        "Autor": "edgar allan poe",
        "Título": "Los crímenes de la Rue Morgue",
        "Precio": 45,
        "FechaCompra": new Date("2019-04-03"),
        "Género": "Hombre",
        "Ciudad": "Buenos Aires",
        "Formación": ["Primaria Completo", "Secundaria Incompleto"]
    }

```

```

    },
    {
      "Código Inventario": "234590",
      "Autor": "Fedor Dostoiewski",
      "Título": "Crimen y castigo",
      "Precio": 40,
      "EdadComprador": 59,
      "Género": "Hombre",
      "Formación": ["Primaria Completo", "Secundaria Incompleto"]
    },
    {
      "Código Inventario": "135426",
      "Autor": "Arthur Conan Doyle",
      "Título": "Las aventuras de Sherlock Holmes",
      "Precio": 29,
      "Género": "F",
      "Ciudad": "Montevideo"
    }
  ]
}

```

_id	Código Inventario	Autor	Título	Precio	FechaCompra	EdadComprador	Ciudad	Formación	Género
64376b31985a069d3c2ff6cf	123321	Alejandro Dumas	Los tres mosqueteros	10	2019-02-01T00:00:00.000Z	80	Lima	[2 elements]	
64376b31985a069d3c2ff6d0	135426	Arthur Conan Doyle	Las aventuras de Sherlock Holmes	11	2019-02-02T00:00:00.000Z	45		[3 elements]	Mujer
64376b31985a069d3c2ff6d1	135426	Arthur Conan Doyle	Las aventuras de Sherlock Holmes	11	2019-03-03T00:00:00.000Z		París		Mujer
64376b31985a069d3c2ff6d2	235678	Charles Dickens	Cuento de Navidad	25		34			
64376b31985a069d3c2ff6d3	987654	Charles Dickens	Historia de dos ciudades	30		57	París		Mujer
64376b31985a069d3c2ff6d4	986532	Charles Dickens	Oliver Twist	12	2019-04-02T00:00:00.000Z			[3 elements]	Mujer
64376b31985a069d3c2ff6d5	876521	Edgar allan Poe	El gato negro	40		27			
64376b31985a069d3c2ff6d6	124567	edgar allan poe	Los crímenes de la Rue Morgue	45	2019-04-03T00:00:00.000Z		Buenos Aires	[2 elements]	Hombre
64376b31985a069d3c2ff6d7	234590	Fedor Dostoiewski	Crimen y castigo	40		59		[2 elements]	Hombre
64376b31985a069d3c2ff6d8	135426	Arthur Conan Doyle	Las aventuras de Sherlock Holmes	29			Montevideo		Mujer

1.2.- Generar una sentencia que permita recuperar los documentos cuyo Autor sea Edgar Allan Poe.

```
db.Ventas.find({ Autor: {$regex: "Edgar Allan Poe", $options: "i"}})
```

```

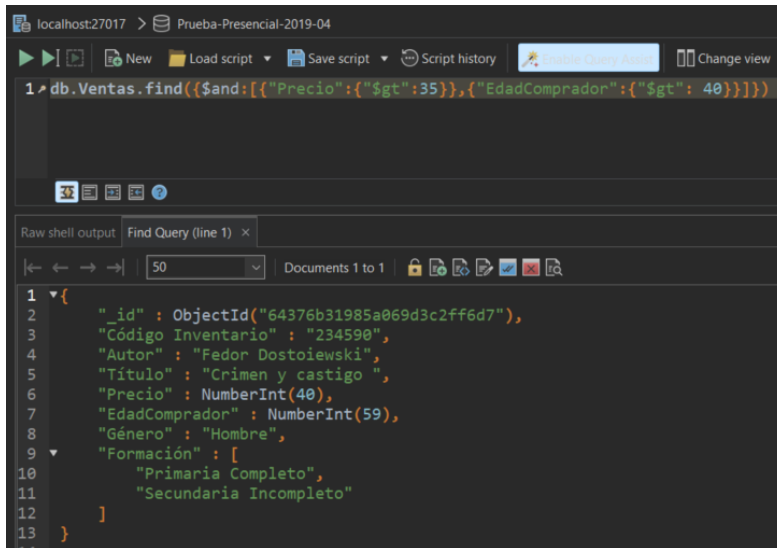
localhost:27017 > Prueba-Presencial-2019-04
1 db.Ventas.find({ Autor: {$regex: "Edgar Allan Poe", $options: "i"}})

Raw shell output Find Query (line 1) x
50 Documents 1 to 2
1 {
2   "id" : ObjectId("64376b31985a069d3c2ff6d5"),
3   "Código Inventario" : "876521",
4   "Autor" : "Edgar allan Poe",
5   "Título" : "El gato negro",
6   "Precio" : NumberInt(40),
7   "EdadComprador" : NumberInt(27)
8 }
9
10 {
11   "id" : ObjectId("64376b31985a069d3c2ff6d6"),
12   "Código Inventario" : "124567",
13   "Autor" : "edgar allan poe",
14   "Título" : "Los crímenes de la Rue Morgue",
15   "Precio" : NumberInt(45),
16   "FechaCompra" : ISODate("2019-04-03T00:00:00.000+0000"),
17   "Género" : "Hombre",
18   "Ciudad" : "Buenos Aires",
19   "Formación" : [
20     "Primaria Completo",
21     "Secundaria Incompleto"
22   ]
23 }

```

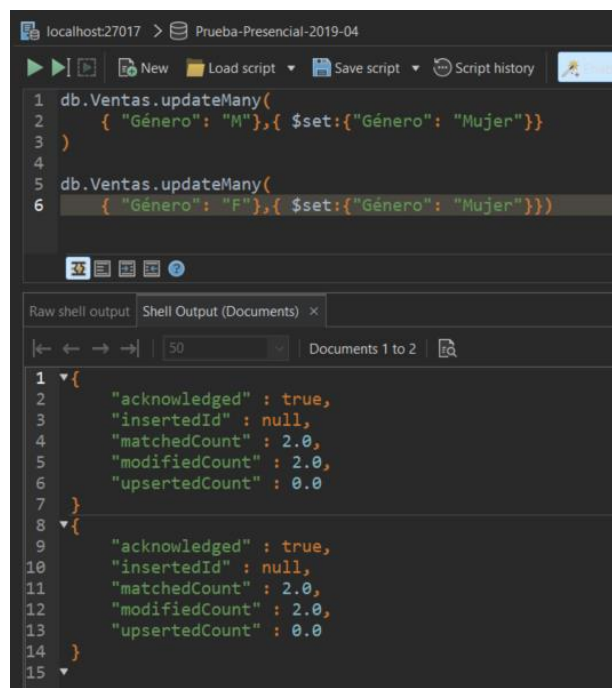
1.3.- Generar una sentencia que permita recuperar los documentos que tengan Precio mayor a 35 y la EdadComprador mayor a 40.

```
db.Ventas.find({$and:[{"Precio":{"$gt":35}},{"EdadComprador":{"$gt":40}}]})
```



1.4.- Unificar los valores de Género, para que queden normalizados (que los valores finales sean Hombre y Mujer).

```
db.Ventas.updateMany(
  {"Género": "M"},{ $set:{"Género": "Mujer"}})
db.Ventas.updateMany(
  {"Género": "F"},{ $set:{"Género": "Mujer"}})
```



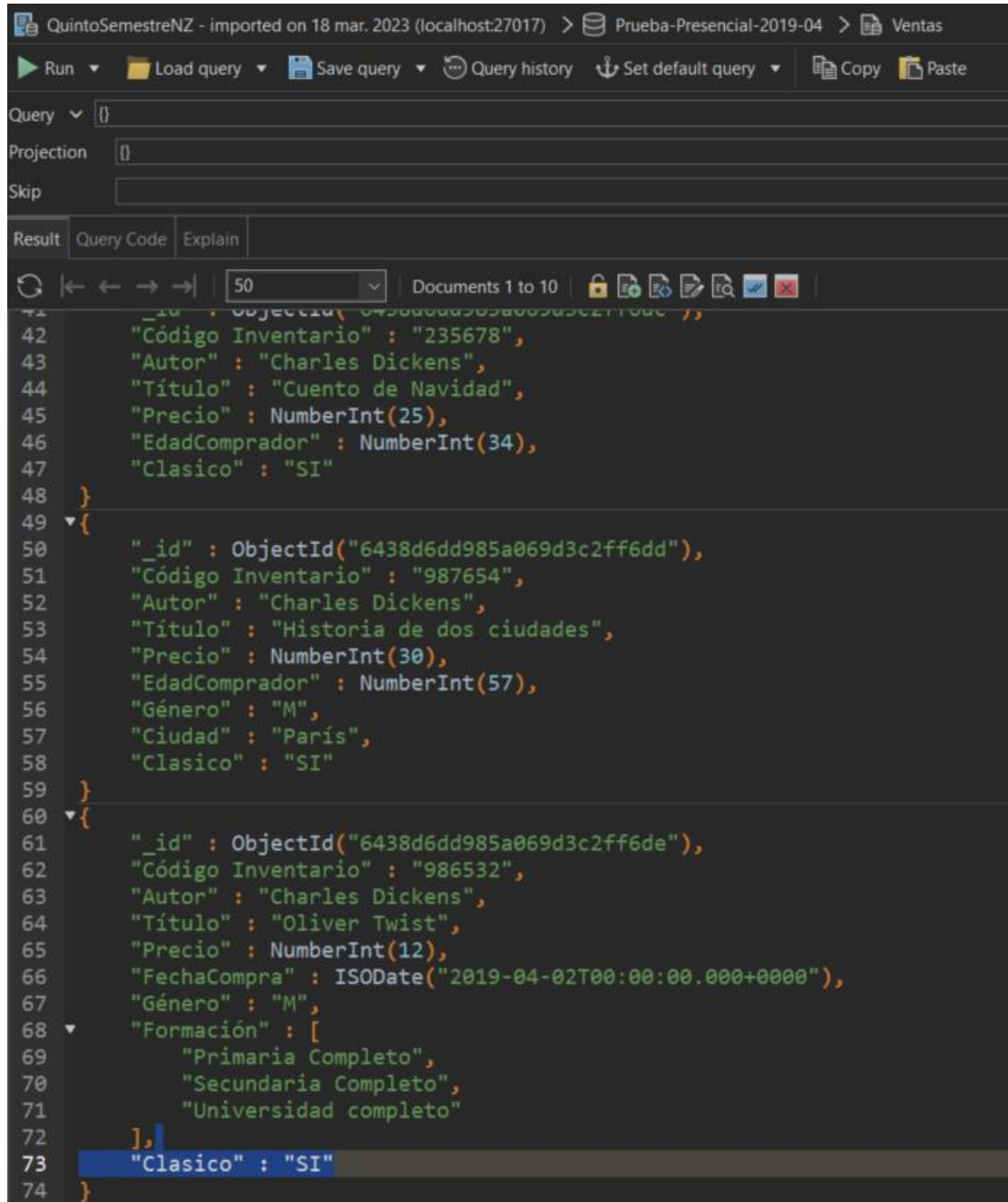
1.5.- Agregar la clave Mes, donde se cargue el mes que se realizó la compra (filtrar para Febrero, Marzo y Abril)

```
db.Ventas.aggregate([  
  $match: { FechaCompra: { $exists: true } }  
],{  
  $addFields: {  
    Mes: {$let: {vars: {month: {$month: "$FechaCompra"}},in:{  
      $switch: {  
        branches: [  
          {case: {$eq: ["$$month", 1]}, then: "Enero"},  
          {case: {$eq: ["$$month", 2]}, then: "Febrero"},  
          {case: {$eq: ["$$month", 3]}, then: "Marzo"},  
          {case: {$eq: ["$$month", 4]}, then: "Abril"},  
          {case: {$eq: ["$$month", 5]}, then: "Mayo"},  
          {case: {$eq: ["$$month", 6]}, then: "Junio"},  
          {case: {$eq: ["$$month", 7]}, then: "Julio"},  
          {case: {$eq: ["$$month", 8]}, then: "Agosto"},  
          {case: {$eq: ["$$month", 9]}, then: "Septiembre"},  
          {case: {$eq: ["$$month", 10]}, then: "Octubre"},  
          {case: {$eq: ["$$month", 11]}, then: "Noviembre"},  
          {case: {$eq: ["$$month", 12]}, then: "Diciembre"}],  
        default: "Mes Inválido"  
      }  
    }  
  }  
}]
```

```
localhost:27017 > Prueba-Preencial-2019-04  
4 $addFields: {  
5   Mes: {$let: {vars: {month: {$month: "$FechaCompra"}  
6   },in: {  
7     $switch: {  
8       branches: [  
9         { case: { $eq: ["$$month", 1] }, then: "Enero" },  
10        { case: { $eq: ["$$month", 2] }, then: "Febrero" },  
11        { case: { $eq: ["$$month", 3] }, then: "Marzo" },  
12        { case: { $eq: ["$$month", 4] }, then: "Abril" },  
13        { case: { $eq: ["$$month", 5] }, then: "Mayo" },  
14        { case: { $eq: ["$$month", 6] }, then: "Junio" },  
15        { case: { $eq: ["$$month", 7] }, then: "Julio" },  
16        { case: { $eq: ["$$month", 8] }, then: "Agosto" },  
17        { case: { $eq: ["$$month", 9] }, then: "Septiembre" },  
18        { case: { $eq: ["$$month", 10] }, then: "Octubre" },  
19        { case: { $eq: ["$$month", 11] }, then: "Noviembre" },  
20        { case: { $eq: ["$$month", 12] }, then: "Diciembre" },  
21        default: "Mes Inválido"  
22      ]  
23    }  
24  }  
25 }  
26 }  
27 }  
28 }  
29 }  
30 }  
31 }  
32 }  
33 }  
34 }  
35 }  
36 }  
37 }  
38 }  
39 }  
40 }  
41 }  
42 }  
43 }  
44 }  
45 }  
46 }  
47 }  
48 }  
49 }  
50 }  
51 }  
52 }  
53 }  
54 }  
55 }  
56 }  
57 }  
58 }  
59 }  
60 }  
61 }  
62 }  
63 }  
64 }  
65 }  
66 }  
67 }  
68 }  
69 }  
70 }  
71 }  
72 }  
73 }  
74 }  
75 }  
76 }  
77 }  
78 }  
79 }  
80 }  
81 }  
82 }  
83 }  
84 }  
85 }  
86 }  
87 }  
88 }  
89 }  
90 }  
91 }  
92 }  
93 }  
94 }  
95 }  
96 }  
97 }  
98 }  
99 }  
100 }  
101 }  
102 }  
103 }  
104 }  
105 }  
106 }  
107 }  
108 }  
109 }  
110 }  
111 }  
112 }  
113 }  
114 }  
115 }  
116 }  
117 }  
118 }  
119 }  
120 }  
121 }  
122 }  
123 }  
124 }  
125 }  
126 }  
127 }  
128 }  
129 }  
130 }  
131 }  
132 }  
133 }  
134 }  
135 }  
136 }  
137 }  
138 }  
139 }  
140 }  
141 }  
142 }  
143 }  
144 }  
145 }  
146 }  
147 }  
148 }  
149 }  
150 }  
151 }  
152 }  
153 }  
154 }  
155 }  
156 }  
157 }  
158 }  
159 }  
160 }  
161 }  
162 }  
163 }  
164 }  
165 }  
166 }  
167 }  
168 }  
169 }  
170 }  
171 }  
172 }  
173 }  
174 }  
175 }  
176 }  
177 }  
178 }  
179 }  
180 }  
181 }  
182 }  
183 }  
184 }  
185 }  
186 }  
187 }  
188 }  
189 }  
190 }  
191 }  
192 }  
193 }  
194 }  
195 }  
196 }  
197 }  
198 }  
199 }  
200 }  
201 }  
202 }  
203 }  
204 }  
205 }  
206 }  
207 }  
208 }  
209 }  
210 }  
211 }  
212 }  
213 }  
214 }  
215 }  
216 }  
217 }  
218 }  
219 }  
220 }  
221 }  
222 }  
223 }  
224 }  
225 }  
226 }  
227 }  
228 }  
229 }  
230 }  
231 }  
232 }  
233 }  
234 }  
235 }  
236 }  
237 }  
238 }  
239 }  
240 }  
241 }  
242 }  
243 }  
244 }  
245 }  
246 }  
247 }  
248 }  
249 }  
250 }  
251 }  
252 }  
253 }  
254 }  
255 }  
256 }  
257 }  
258 }  
259 }  
260 }  
261 }  
262 }  
263 }  
264 }  
265 }  
266 }  
267 }  
268 }  
269 }  
270 }  
271 }  
272 }  
273 }  
274 }  
275 }  
276 }  
277 }  
278 }  
279 }  
280 }  
281 }  
282 }  
283 }  
284 }  
285 }  
286 }  
287 }  
288 }  
289 }  
290 }  
291 }  
292 }  
293 }  
294 }  
295 }  
296 }  
297 }  
298 }  
299 }  
300 }  
301 }  
302 }  
303 }  
304 }  
305 }  
306 }  
307 }  
308 }  
309 }  
310 }  
311 }  
312 }  
313 }  
314 }  
315 }  
316 }  
317 }  
318 }  
319 }  
320 }  
321 }  
322 }  
323 }  
324 }  
325 }  
326 }  
327 }  
328 }  
329 }  
330 }  
331 }  
332 }  
333 }  
334 }  
335 }  
336 }  
337 }  
338 }  
339 }  
340 }  
341 }  
342 }  
343 }  
344 }  
345 }  
346 }  
347 }  
348 }  
349 }  
350 }  
351 }  
352 }  
353 }  
354 }  
355 }  
356 }  
357 }  
358 }  
359 }  
360 }  
361 }  
362 }  
363 }  
364 }  
365 }  
366 }  
367 }  
368 }  
369 }  
370 }  
371 }  
372 }  
373 }  
374 }  
375 }  
376 }  
377 }  
378 }  
379 }  
380 }  
381 }  
382 }  
383 }  
384 }  
385 }  
386 }  
387 }  
388 }  
389 }  
390 }  
391 }  
392 }  
393 }  
394 }  
395 }  
396 }  
397 }  
398 }  
399 }  
400 }  
401 }  
402 }  
403 }  
404 }  
405 }  
406 }  
407 }  
408 }  
409 }  
410 }  
411 }  
412 }  
413 }  
414 }  
415 }  
416 }  
417 }  
418 }  
419 }  
420 }  
421 }  
422 }  
423 }  
424 }  
425 }  
426 }  
427 }  
428 }  
429 }  
430 }  
431 }  
432 }  
433 }  
434 }  
435 }  
436 }  
437 }  
438 }  
439 }  
440 }  
441 }  
442 }  
443 }  
444 }  
445 }  
446 }  
447 }  
448 }  
449 }  
450 }  
451 }  
452 }  
453 }  
454 }  
455 }  
456 }  
457 }  
458 }  
459 }  
460 }  
461 }  
462 }  
463 }  
464 }  
465 }  
466 }  
467 }  
468 }  
469 }  
470 }  
471 }  
472 }  
473 }  
474 }  
475 }  
476 }  
477 }  
478 }  
479 }  
480 }  
481 }  
482 }  
483 }  
484 }  
485 }  
486 }  
487 }  
488 }  
489 }  
490 }  
491 }  
492 }  
493 }  
494 }  
495 }  
496 }  
497 }  
498 }  
499 }  
500 }  
501 }  
502 }  
503 }  
504 }  
505 }  
506 }  
507 }  
508 }  
509 }  
510 }  
511 }  
512 }  
513 }  
514 }  
515 }  
516 }  
517 }  
518 }  
519 }  
520 }  
521 }  
522 }  
523 }  
524 }  
525 }  
526 }  
527 }  
528 }  
529 }  
530 }  
531 }  
532 }  
533 }  
534 }  
535 }  
536 }  
537 }  
538 }  
539 }  
540 }  
541 }  
542 }  
543 }  
544 }  
545 }  
546 }  
547 }  
548 }  
549 }  
550 }  
551 }  
552 }  
553 }  
554 }  
555 }  
556 }  
557 }  
558 }  
559 }  
560 }  
561 }  
562 }  
563 }  
564 }  
565 }  
566 }  
567 }  
568 }  
569 }  
570 }  
571 }  
572 }  
573 }  
574 }  
575 }  
576 }  
577 }  
578 }  
579 }  
580 }  
581 }  
582 }  
583 }  
584 }  
585 }  
586 }  
587 }  
588 }  
589 }  
590 }  
591 }  
592 }  
593 }  
594 }  
595 }  
596 }  
597 }  
598 }  
599 }  
600 }  
601 }  
602 }  
603 }  
604 }  
605 }  
606 }  
607 }  
608 }  
609 }  
610 }  
611 }  
612 }  
613 }  
614 }  
615 }  
616 }  
617 }  
618 }  
619 }  
620 }  
621 }  
622 }  
623 }  
624 }  
625 }  
626 }  
627 }  
628 }  
629 }  
630 }  
631 }  
632 }  
633 }  
634 }  
635 }  
636 }  
637 }  
638 }  
639 }  
640 }  
641 }  
642 }  
643 }  
644 }  
645 }  
646 }  
647 }  
648 }  
649 }  
650 }  
651 }  
652 }  
653 }  
654 }  
655 }  
656 }  
657 }  
658 }  
659 }  
660 }  
661 }  
662 }  
663 }  
664 }  
665 }  
666 }  
667 }  
668 }  
669 }  
670 }  
671 }  
672 }  
673 }  
674 }  
675 }  
676 }  
677 }  
678 }  
679 }  
680 }  
681 }  
682 }  
683 }  
684 }  
685 }  
686 }  
687 }  
688 }  
689 }  
690 }  
691 }  
692 }  
693 }  
694 }  
695 }  
696 }  
697 }  
698 }  
699 }  
700 }  
701 }  
702 }  
703 }  
704 }  
705 }  
706 }  
707 }  
708 }  
709 }  
710 }  
711 }  
712 }  
713 }  
714 }  
715 }  
716 }  
717 }  
718 }  
719 }  
720 }  
721 }  
722 }  
723 }  
724 }  
725 }  
726 }  
727 }  
728 }  
729 }  
730 }  
731 }  
732 }  
733 }  
734 }  
735 }  
736 }  
737 }  
738 }  
739 }  
740 }  
741 }  
742 }  
743 }  
744 }  
745 }  
746 }  
747 }  
748 }  
749 }  
750 }  
751 }  
752 }  
753 }  
754 }  
755 }  
756 }  
757 }  
758 }  
759 }  
760 }  
761 }  
762 }  
763 }  
764 }  
765 }  
766 }  
767 }  
768 }  
769 }  
770 }  
771 }  
772 }  
773 }  
774 }  
775 }  
776 }  
777 }  
778 }  
779 }  
780 }  
781 }  
782 }  
783 }  
784 }  
785 }  
786 }  
787 }  
788 }  
789 }  
790 }  
791 }  
792 }  
793 }  
794 }  
795 }  
796 }  
797 }  
798 }  
799 }  
800 }  
801 }  
802 }  
803 }  
804 }  
805 }  
806 }  
807 }  
808 }  
809 }  
810 }  
811 }  
812 }  
813 }  
814 }  
815 }  
816 }  
817 }  
818 }  
819 }  
820 }  
821 }  
822 }  
823 }  
824 }  
825 }  
826 }  
827 }  
828 }  
829 }  
830 }  
831 }  
832 }  
833 }  
834 }  
835 }  
836 }  
837 }  
838 }  
839 }  
840 }  
841 }  
842 }  
843 }  
844 }  
845 }  
846 }  
847 }  
848 }  
849 }  
850 }  
851 }  
852 }  
853 }  
854 }  
855 }  
856 }  
857 }  
858 }  
859 }  
860 }  
861 }  
862 }  
863 }  
864 }  
865 }  
866 }  
867 }  
868 }  
869 }  
870 }  
871 }  
872 }  
873 }  
874 }  
875 }  
876 }  
877 }  
878 }  
879 }  
880 }  
881 }  
882 }  
883 }  
884 }  
885 }  
886 }  
887 }  
888 }  
889 }  
890 }  
891 }  
892 }  
893 }  
894 }  
895 }  
896 }  
897 }  
898 }  
899 }  
900 }  
901 }  
902 }  
903 }  
904 }  
905 }  
906 }  
907 }  
908 }  
909 }  
910 }  
911 }  
912 }  
913 }  
914 }  
915 }  
916 }  
917 }  
918 }  
919 }  
920 }  
921 }  
922 }  
923 }  
924 }  
925 }  
926 }  
927 }  
928 }  
929 }  
930 }  
931 }  
932 }  
933 }  
934 }  
935 }  
936 }  
937 }  
938 }  
939 }  
940 }  
941 }  
942 }  
943 }  
944 }  
945 }  
946 }  
947 }  
948 }  
949 }  
950 }  
951 }  
952 }  
953 }  
954 }  
955 }  
956 }  
957 }  
958 }  
959 }  
960 }  
961 }  
962 }  
963 }  
964 }  
965 }  
966 }  
967 }  
968 }  
969 }  
970 }  
971 }  
972 }  
973 }  
974 }  
975 }  
976 }  
977 }  
978 }  
979 }  
980 }  
981 }  
982 }  
983 }  
984 }  
985 }  
986 }  
987 }  
988 }  
989 }  
990 }  
991 }  
992 }  
993 }  
994 }  
995 }  
996 }  
997 }  
998 }  
999 }  
1000 }
```


1.6.- Agregar la clave Clasico, donde se le cargue el valor SI cuando el Autor es Dumas o Dickens.

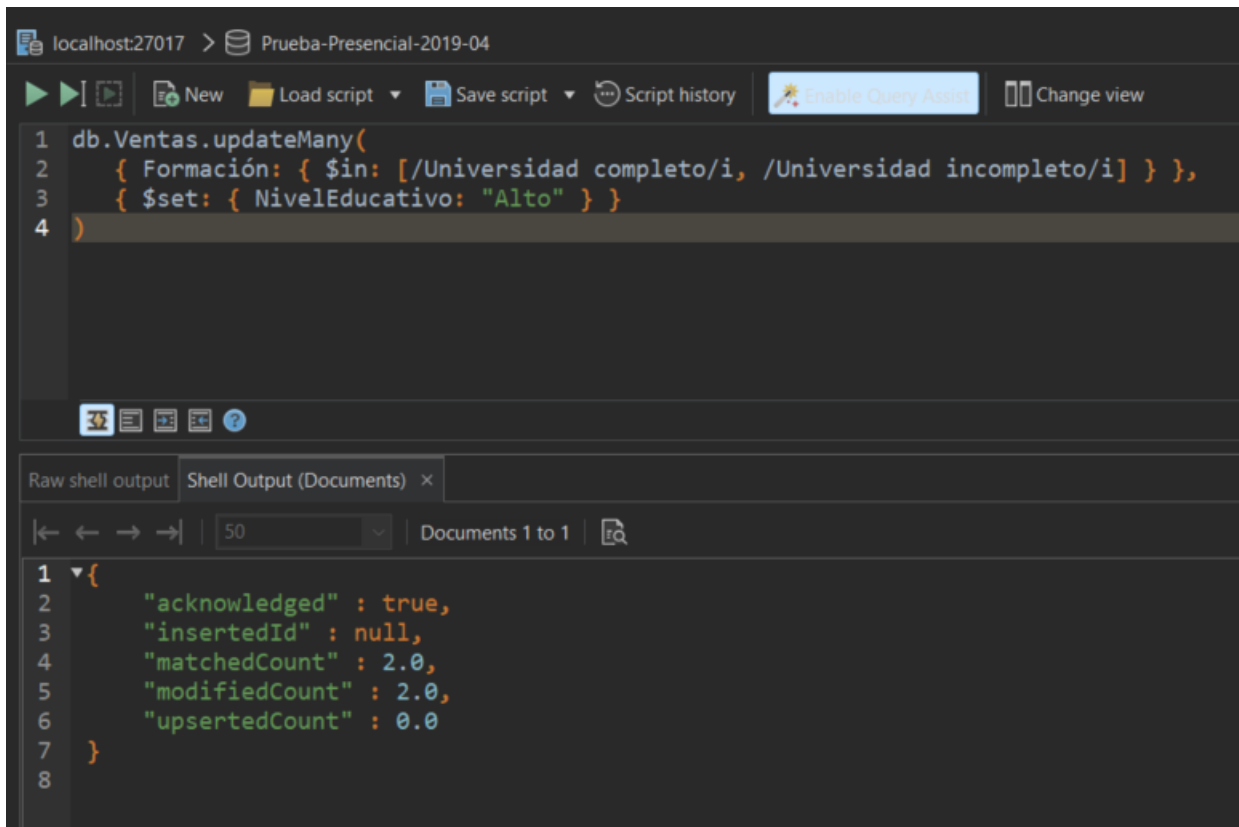
```
db.Ventas.updateMany({Autor: {$regex: "Dickens" , $options: "m"}} ,  
{$set: {Clasico: "SI"}})
```



```
QuintoSemestreNZ - imported on 18 mar. 2023 (localhost:27017) > Prueba-Presencial-2019-04 > Ventas  
Run Load query Save query Query history Set default query Copy Paste  
Query {}  
Projection {}  
Skip  
Result Query Code Explain  
50 Documents 1 to 10  
41 {  
42   "Codigo Inventario" : "235678",  
43   "Autor" : "Charles Dickens",  
44   "Titulo" : "Cuento de Navidad",  
45   "Precio" : NumberInt(25),  
46   "EdadComprador" : NumberInt(34),  
47   "Clasico" : "SI"  
48 }  
49 {  
50   "_id" : ObjectId("6438d6dd985a069d3c2ff6dd"),  
51   "Codigo Inventario" : "987654",  
52   "Autor" : "Charles Dickens",  
53   "Titulo" : "Historia de dos ciudades",  
54   "Precio" : NumberInt(30),  
55   "EdadComprador" : NumberInt(57),  
56   "Genero" : "M",  
57   "Ciudad" : "París",  
58   "Clasico" : "SI"  
59 }  
60 {  
61   "_id" : ObjectId("6438d6dd985a069d3c2ff6de"),  
62   "Codigo Inventario" : "986532",  
63   "Autor" : "Charles Dickens",  
64   "Titulo" : "Oliver Twist",  
65   "Precio" : NumberInt(12),  
66   "FechaCompra" : ISODate("2019-04-02T00:00:00.000+0000"),  
67   "Genero" : "M",  
68   "Formación" : [  
69     "Primaria Completo",  
70     "Secundaria Completo",  
71     "Universidad completo"  
72   ],  
73   "Clasico" : "SI"  
74 }
```

1.7.- Agregar la clave NivelEducativo, donde se le cargue el valor Alto si tiene formación Universitaria (completa o incompleta).

```
db.Ventas.updateMany(  
  {Formación: {$in: [/Universidad completo/i, /Universidad  
incompleto/i]}},  
  {$set: {NivelEducativo: "Alto"}})
```



The screenshot shows a MongoDB Shell interface with the following components:

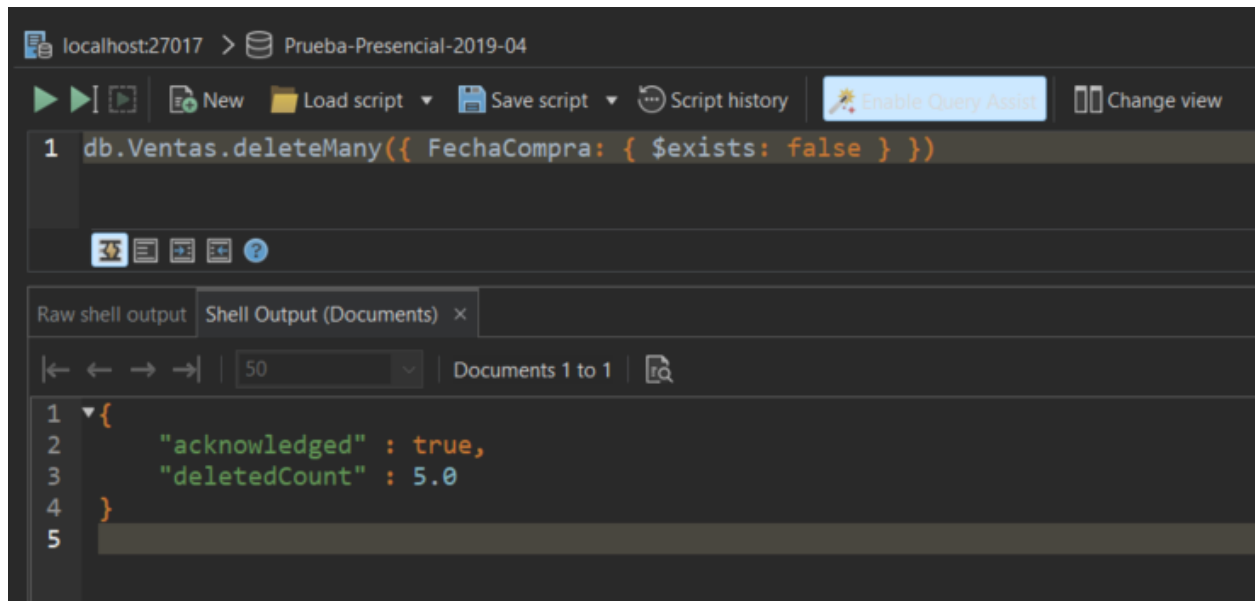
- Header:** localhost:27017 > Prueba-Presencial-2019-04
- Toolbar:** Includes icons for running, saving, and loading scripts, along with buttons for "New", "Load script", "Save script", "Script history", "Enable Query Assist", and "Change view".
- Script Editor:** Contains the following JavaScript code:

```
1 db.Ventas.updateMany(  
2   { Formación: { $in: [/Universidad completo/i, /Universidad  
3     incompleto/i] } },  
4   { $set: { NivelEducativo: "Alto" } }  
5 )
```
- Output Panel:** Shows the "Shell Output (Documents)" tab with the following JSON response:

```
1 {  
2   "acknowledged" : true,  
3   "insertedId" : null,  
4   "matchedCount" : 2.0,  
5   "modifiedCount" : 2.0,  
6   "upsertedCount" : 0.0  
7 }  
8
```

1.8.- Eliminar los documentos que no tengan clave FechaCompra

```
db.Ventas.deleteMany({ FechaCompra: { $exists: false } })
```



The screenshot shows a MongoDB Shell interface. The top bar indicates the connection is to 'localhost:27017' and the database is 'Prueba-Presencial-2019-04'. The command editor contains the following command:

```
1 db.Ventas.deleteMany({ FechaCompra: { $exists: false } })
```

Below the command editor, the 'Shell Output (Documents)' tab is active, showing the result of the command:

```
1 {
2   "acknowledged" : true,
3   "deletedCount" : 5.0
4 }
5
```

2.- Ejercicios Java:

- 2.1.- Crear la colección VentasJava que registre los documentos de la planilla Datos.xlsx. Los datos deben cargarse como en la planilla, respetando las mayúsculas y minúsculas y solo cargándoles la clave cuando existen valor. La clave Formación es de tipo array.**
- 2.2.- Generar una sentencia que permita recuperar los documentos cuyo Autor sea Edgar Allan Poe.**
- 2.3.- Generar una sentencia que permita recuperar los documentos que tengan Precio mayor a 35 y la EdadComprador mayor a 40.**
- 2.4.- Unificar los valores de Género, para que queden normalizados (que los valores finales sean Hombre y Mujer).**
- 2.5.- Agregar la clave Mes, donde se cargue el mes que se realizó la compra (filtrar para Febrero, Marzo y Abril).**
- 2.6.- Agregar la clave Clasico, donde se le cargue el valor SI cuando el Autor es Dumas o Dickens.**
- 2.7.- Agregar la clave NivelEducativo, donde se le cargue el valor Alto si tiene formación Universitaria (completa o incompleta).**

2.8.- Eliminar los documentos que no tengan clave FechaCompra

3.- Parte teórica.

1.- Cuando hablamos de Consistencia Eventual, hay dos maneras de garantizar que eventualmente los datos serán consistentes:

- a) Basic Availability y Delayed Repair
- b) Delayed Repair y Read Repair**
- c) Delayed Repair y Soft State
- d) Read Repair y Soft State

Para garantizar que eventualmente los datos serán consistentes, se utilizan dos técnicas:

- Delayed Repair: La reparación o reconciliación de los datos inconsistentes se realiza de manera asíncrona y se permite cierto retraso en este proceso.
- Read Repair: Cuando se detecta que una réplica tiene datos inconsistentes con otras, se actualiza la réplica con los datos correctos durante una lectura de los datos.

Por lo tanto, la opción b, que incluye Delayed Repair y Read Repair, es la respuesta correcta.

2.- Las bases de datos multi modelo tienen las siguientes características:

- a) Almacena los datos como documentos estructurados, generalmente XML o formatos JSON.
- b) Cada elemento en la base de datos se almacena como el nombre de un atributo, o clave, junto con su valor.
- c) Ofrece las ventajas de la persistencia políglota sin sus desventajas.**
- d) La esencia de este modelo es buscar una estructura que permita que los datos sean almacenados en disco de forma consecutiva, optimizando la búsqueda de datos, y esto se resuelve si los datos son almacenados en columnas, en vez de filas.

3.- ¿Cuál de estas sentencias elimina solo un documento de la colección Libros, que cumplen con la condición de que el Autor es Juan?

- a) `db.libros.updateOne({autor: "Juan"}, {$set: null})`
- b) `db.libros.removeOne({})`
- c) `db.libros.removeMany({autor: "Juan"})`
- d) `db.libros.removeOne({autor: "Juan"})`

4.- MongoDB ofrece funciones de agregación que permiten procesar datos y devolver resultados calculados. Las cuatro formas de agregación que se ofrecen son:

1. aggregation pipeline
2. map-reduce
3. métodos simples de agregación
4. joins

Esto es Falso.

Agregación.

Las funciones de agregación permiten procesar datos y devolver resultados calculados. Básicamente agrupan los valores de varios documentos, y se pueden realizar operaciones sobre esos documentos agrupados, para devolver un solo resultado.

Existen tres formas de agregación:

- aggregation pipeline
- map-reduce
- métodos simples de agregación