

Guia # 5 – MongoDB

(**Conectores Lógicos**)

12.- CONECTIVOS LÓGICOS

12.1.- **\$and: [{P1}, {P2}, ..., {Pn}]**

Ejemplo 1: Mostrar el id, el nombre, la edad y el género solo de los caballeros con 37 años

```
> db.estudiantes.find( { $and:[  
    { genero:{ $eq:"m" } },  
    { edad:{ $eq:37 } }  
] },  
    {  
        _id:1, nombre:1,  
        edad:1,  
        genero:1  
    }  
).toArray() ;
```

21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44

Ejemplo 2: Mostrar el id, el nombre, el género, la edad y los programas que manejan las damas con edad entre 23 (Inclusive) y 25 años (Inclusive)

```
> db.estudiantes.find( {  
    $and:[  
        { genero:{ $eq:"f" } } ,  
        { edad:{ $gte: 23 } } ,  
        { edad:{ $lte : 30 } }  
    ]  
    },  
    {  
        _id:1,  
        nombre:1,  
        genero:1,  
        edad:1,  
        programa:1  
    }  
).toArray() ;
```

45

46

47 **Ejemplo 3:** Mostrar el id, el nombre, el genero , la edad y las materias

48 Que cursan las damas con la nota de programación 1 >= 4.0 y edad >= 50

49 > db.estudiantes.find({

50 **\$and:[**

51 {genero:{ \$eq:"f" }} ,

52 {"materias.programacion1":{ \$gte:4.0 }} ,

53 {edad:{ \$gte: 50 }}

54]

55 },

56 {

57 _id:1,

58 nombre:1,

59 genero:1,

60 edad:1,

61 materias:1

62 }

63).toArray() ;

64

65 > db.estudiantes.find({ \$and:[{genero:{ \$eq:"f" }} ,

66 {"materias.programacion1":{ \$gte:4.0}} , {edad:{ \$gte: 50 }}] },

67 { _id:1, nombre:1, edad:1, genero:1, materias.programacion1":1}).toArray();

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

Ejemplo 4: Mostrar el id, el nombre, el genero y los programas que manejan las damas que saben programar en Python

```
> db.estudiantes.find({
    $and:[
        { programa: { $eq:"Python"} } ,
        { genero: { $eq:"f" } }
    ]
},
{
    _id:1,
    nombre:1,
    genero:1,
    programa:1
}).toArray();
```

95

96

97 **Ejemplo 5:** Mostrar el id, el nombre, el genero y los programas que manejan
98 las damas que saben programar en Python y Java

99

100 > db.estudiantes.find({

101

\$and:[

102

{ programa: { \$eq:"Python" } } ,

103

{ programa: { \$eq:"Java" } } ,

104

{ genero: { \$eq:"f" } }

105

]

106

} ,

107

{

108

_id:1,

109

nombre:1,

110

genero:1,

111

programa:1

112

}

113

).toArray();

114

115

116

117

118

119

120

121

122 **12.2.- \$OR: [{P1}, {P2}, ..., {Pn}]**

123 **Ejemplo 1:** Mostrar el id, el nombre, la edad y el género, de las personas que tengan
124 género masculino o edad igual a 37

125

126

127 `> db.estudiantes.find({`

128 `$or:[`

129 `{ genero: { $eq:"m" } } ,`

130 `{ edad: { $eq:37 } } ,`

131 `]`

132 `},`

133 `{`

134 `_id:1,`

135 `nombre:1,`

136 `edad:1,`

137 `genero:1,`

138 `}`

139 `).toArray();`

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

Ejemplo 2: Mostrar el id, el nombre, la edad y el género, de las personas que tengan como edad 24 o 37

```
➤ db.estudiantes.find(  
    $or:[  
        { edad: { $eq:24} } ,  
        { edad: { $eq:37} }  
    ]  
    ,  
    {  
        _id:1,  
        nombre:1,  
        edad:1,  
        genero:1,  
    }  
).toArray();
```

169

170

171 **Ejemplo 3:** Se desea mostrar de los estudiantes que dominan los programas Python o
172 Java: el id, el nombre, el género y los programas que manejan los estudiantes

173

174 > db.estudiantes.find({

175 **\$or:[**

176 { programa: { \$eq: "Python" } } ,

177 { programa: { \$eq: "Java" } } }

178]

179 } ,

180 {

181 _id:1,

182 nombre:1,

183 genero:1,

184 programa:1

185 }

186).toArray();

187

188

189

190 > db.estudiantes.find({ \$or: [{ programa: "Python" } , { programa: "Java" }] },

191 { _id:1, nombre:1, edad:1, programa:1 }).toArray();

192

193

194

195
196

197 **12.3.- \$not: { Pregunta }**

198 **Ejemplo 1:** Ejemplo 1: Se desea un listado de los estudiantes cuyo genero no sea
199 masculino.

200

```
201 > db.estudiantes.find(  
202     genero: {  
203         $not:{ $eq: "m" }  
204     }  
205 },  
206     {  
207         _id:1,  
208         nombre:1,  
209         edad:1,  
210         genero:1,  
211         estado:1  
212     }  
213 ).toArray();
```

214
215
216
217
218
219
220

221

222 **12.4.- Conectores anidados**

223

224 **Ejemplo 1:** Se desea mostrar los datos de las damas que tiene como edad 24 o 37

225 dbuniversidad> db.estudiantes.find({

226 **\$and:[**

227 **{** genero: **{ \$eq:"f" }** **},**

228 **{**

229 **\$or:[**

230 **{** edad **:{ \$eq:37 }** **},**

231 **{**edad **:{ \$eq:24 }** **}**

232 **]**

233 **}**

234 **]**

235 **},**

236 **{**

237 **_id:1,** nombre:1,

238 **edad:1,** genero:1

239 **}**

240 **).toArray();**

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

Ejemplo 2: Se desea mostrar los datos de los estudiantes que no son caballeros y no tengan una edad <= 40

```
> db.estudiantes.find(  
    $and:[  
        { genero: { $not: { $eq: "m" } } } ,  
        { edad: { $not: { $lte: 40 } } }  
    ]  
    } ,  
    {  
        _id:1,  
        nombre:1,  
        edad:1,  
        genero:1  
    }  
).toArray()
```

268

269

270 **Ejemplo 3:** Ejemplo 1: Mostrar los datos de las personas que el estado civil no sea
271 casada o viuda.

272 > db.estudiantes.find({

273 **estado: {**

274 **\$not:{**

275 **\$in: [**

276 **"Casada",**

277 **"Casado",**

278 **"Viuda",**

279 **"Viudo"**

280 **]**

281 **}**

282 **}**

283 **} ,**

284 {

285 **_id:1, nombre:1, edad:1,**

286 **genero:1, estado:1**

287 }

288).toArray();

289

290