Realice la instrucción para dar respuesta a las siguientes consultas:

Habilite el archivo para escritura para que de cada sala coloque una consulta.

1. Mostrar la información completa de los amigos de "Wilbur Swindles”

cant:1

| filter | {first\_name:'Wilbur',last\_name:'Swindles'} |
| --- | --- |
| project | {friends:1} |
| sort |  |

1. Mostrar los estudiantes que están en la clase 6 y son de género masculino.

cant:13

| filter | {class: 6, gender:"Male"} |
| --- | --- |
| project |  |
| sort |  |

1. Encuentre los nombres, apellidos y clase de los estudiantes, que tienen algún amigo cuyo número de la suerte es el 13, ordenados por grado en orden ascendente. (No mostrar el \_id)

cant:6

| filter | {"friends.luckyNumber":13} |
| --- | --- |
| project | {first\_name:1,last\_name:1,class:1,\_id:0} |
| sort | {class:1} |

1. Clases en las que hay mujeres que aman las matemáticas, sin mostrar el campo \_id (se pueden mostrar valores duplicados).

Cant:45

| filter | {loveMaths:true,gender:'Female'} |
| --- | --- |
| project | {first\_name:1,gender:1,loveMaths:1,\_id:0} |

db.students.find({loveMaths:true,gender:'Female'},{first\_name:1,gender:1,loveMaths:1,\_id:0});

1. Listado de nombre y apellido de los estudiantes hombres que están en la clase 2 o 5, ordenados alfabéticamente por nombre y apellido.

Cant: 23

| filter | {$or:[{class:2},{class:5}],gender:"Male"} |
| --- | --- |
| project |  |
| sort | {first\_name:1,last\_name:1} |

1. Encuentre el nombre, apellido, clase y género de los estudiantes, que tengan entre sus ciudades favoritas a Roma o alguna que inicie con “Paris”.

Cant: 3

| filter | {$or:[{favCity:"Roma"},{favCity:/^Paris/}]} |
| --- | --- |
| project | {first\_name:1, last\_name:1, class:1, gender:1} |
| sort |  |

1. Listado de las mujeres que tienen como mejores amigas a solo mujeres.

cant: 77

| filter | {"friends.gender": 'Female', gender: 'Female'} |
| --- | --- |
| project | { \_id: 0} |
| sort |  |

db.students.find({"friends.gender": 'Female', gender: 'Female'},{ \_id: 0})

1. Cantidad de estudiantes que aman y que no aman las matemáticas.

{\_id:"$loveMaths",

count:{$sum:1}

}

True: 105

False: 98

1. Mostrar las clases en las que hay estudiantes, sin duplicados.

| distinct | db.students.distinct("class"); |
| --- | --- |

1. Contar por clase, cuántos estudiantes hay.

| $group | { \_id: "$class",  num: {$sum:1}  } |
| --- | --- |

Otras consultas:

1. Cantidad de estudiantes que tienen amigos con un número de la suerte mayor a 50
2. Número de la clase que tiene más cantidad de estudiantes.
3. Contar la cantidad de amigos que hay por cada número de la suerte, ordenarlo de mayor a menor cantidad.