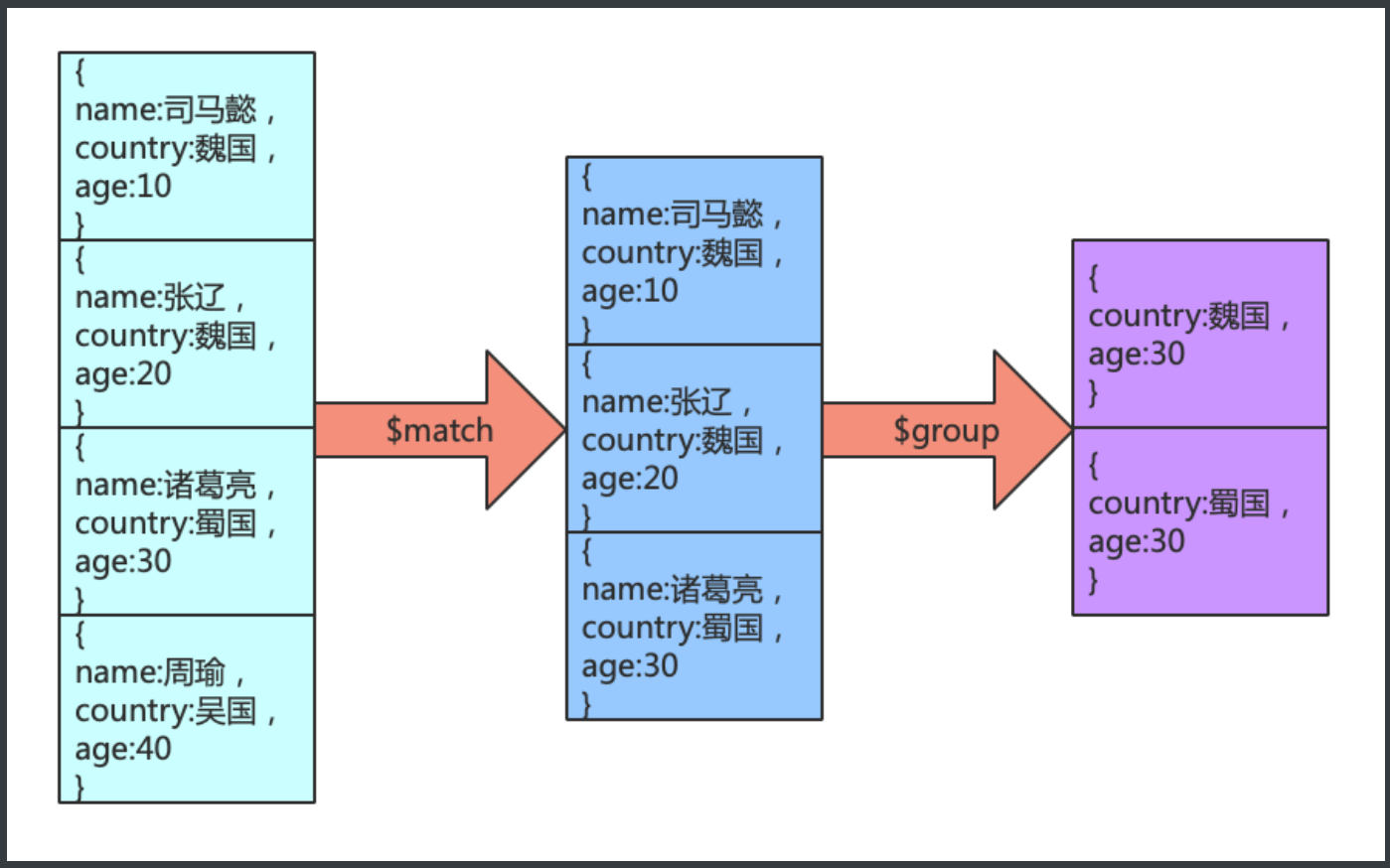
1 mongodb的聚合是什么

MongoDB 中聚合(aggregate)主要用于处理数据(诸如统计平均值，求和等)，并返回计算后的数据结果。

有点类似 **SQL** 语句中的 **count(*)**。

语法： `db.集合名称.aggregate([{管道:{表达式}}])`



2 管道命令之 $group

2.1 按照某个字段进行分组

`$group` 是所有聚合命令中用的最多的一个命令，用来将集合中的文档分组，可用于统计结果

使用示例如下

```
db.stu.aggregate(
    {$group:
        {
            _id:"$country",
            counter:{$sum:1}
        }
    }
)
```

其中注意点：

- `db.db_name.aggregate` 是语法，所有的管道命令都需要写在其中
- `_id` 表示分组的依据，按照哪个字段进行分组，需要使用 `$gender` 表示选择这个字段进行分组
- `$sum:1` 表示把每条数据作为1进行统计，统计的是该分组下面数据的条数

2.2 常用表达式

表达式：处理输入文档并输出 语法： `表达式:'$列名'` 常用表达式:

- `$sum`： 计算总和， $sum:1 表示以一倍计数
- `$avg`： 计算平均值
- `$min`： 获取最小值
- `$max`： 获取最大值
- `$push`： 在结果文档中插入值到一个数组中

3 管道命令之 $match

`$match` 用于进行数据的过滤，是在能够在聚合操作中使用的命令，和 `find` 区别在于 `$match` 操作可以把结果交给下一个管道处理，而 `find` 不行

使用示例如下：

1. 查询年龄大于20的人

```
db.person.aggregate([
    {$match:{age:{$gt:20}}}
    ])
```

2. 查询年龄大于20的魏国的人数

```
db.person.aggregate([
    {$match:{age:{$gt:20}}},
    {$group:{_id:"$country",counter:{$sum:1}}}
    ])
```

4 管道命令之 $project

`$project` 用于修改文档的输入输出结构，例如重命名，增加，删除字段

使用示例如下：

1. 查询人物的姓名、年龄，不显示ID

```
db.person.aggregate([
    {$project:{_id:0,name:1,age:1}}
    ])
```

2. 查询每个国家的人数，只显示数量

```
db.person.aggregate([
    {$group:{_id:"$country",counter:{$sum:1}}},
    {$project:{_id:0,counter:1}}
    ])
```

6 管道命令之 $sort

`$sort` 用于将输入的文档排序后输出

使用示例如下：

1. 查询人物，按照年龄升序

```
db.person.aggregate([{$sort:{age:1}}])
```

2. 查询每个国家的人数，并排序

```
db.person.aggregate([
    {$group:{_id:"$country",counter:{$sum:1}}},
    {$sort:{counter:-1}}
])
```

7 管道命令之 $skip 和 $limit

- `$limit` 限制返回数据的条数
- `$skip` 跳过指定的文档数，并返回剩下的文档数
- 同时使用时先使用skip在使用limit

使用示例如下：

1. 查询2条信息

```
db.person.aggregate([
    {$limit:2}
])
```

2. 查询从第三条开始的信息

```
db.person.aggregate([
    {$skip:3}
])
```

3. 查询每个国家的人数，按照人数升序，返回第二条数据

```
db.person.aggregate([
    {$group:{_id:"$country",counter:{$sum:1}}},
    {$sort:{counter:-1}},
    {$skip:1},
    {$limit:1}
])
```